University Microfilms International
300 North Zeeb Road
Ann Arbor, Michigan 48106 USA
St. John's Road, Tyler's Green
High Wycombe, Bucks, England HP10 8HR

METHODOLOGY FOR THE AUTOMATION OF THE AUDIT PROCESS INVOLVING THE EVALUATION OF THE PLAN OF INTERNAL CONTROL

by

Arthur Zale Lieberman

A Dissertation Submitted to the Faculty of the

COMMITTEE ON BUSINESS ADMINISTRATION

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

1977

THE UNIVERSITY OF ARIZONA

GRADUATE COLLEGE

I hereby recommend that this dissertation prepared under my direction by Arthur Zale Lieberman

entitled Methodology for the Automation of the Audit Process Involving the Evaluation of the Plan of Internal Control

be accepted as fulfilling the dissertation requirement for the degree of Doctor of Philosophy

July 6, 1977

Dissertation Director — Date

As members of the Final Examination Committee, we certify that we have read this dissertation and agree that it may be presented for final defense.

William B. Barrett — July 7, 1977

B. D. Ripley — July 11, 1977

Mary E. Loomis — July 11, 1977

Final approval and acceptance of this dissertation is contingent on the candidate's adequate performance and defense thereof at the final oral examination.

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED:

ACKNOWLEDGMENTS

I want to thank very much those people who have supported me and contributed to my doctoral research. Dr. Jay F. Nunamaker supported me with insight, encouragement, and friendship throughout my graduate studies at both Purdue University and The University of Arizona, serving as my major advisor. Dr. Mary S. Loomis has been a great friend, provided much encouragement in my efforts, and volunteered invaluable editorial assistance. Dr. Edward S. Lynn, without whose assistance this research would have collapsed, offered a great amount of input on the auditing aspects of the research. Dr. Benn R. Konsynski offered much support in the area of research methodologies. Dr. Andrew B. Whinston at Purdue University taught me to observe, abstract, and create, and I am indebted to him for awakening my mind. Dr. G. David Ripley opened my eyes to the need for concern in programming style, and offered support for the compiler aspects of my research. I thank Dr. William B. Barrett for filling in on committees at two very critical points along the way. Nathan I. Lieberman, C.P.A., and Philip L. Lieberman were invaluable resources as my links to the real worlds of accounting and data processing practices. Mary S. Koch co-authored a paper with me on the effectiveness of currently applied computer auditing techniques, the results of which are included in Chapter 3. Appreciation is extended to C.P.A.'s Lawrence Norrid of The University of Arizona, Elmer Thierman of Peat, Marwick, Mitchell & Co., and Thomas Jones of Ernst & Ernst for offering feedback concerning real-world expectations of my proposed methodology. Finally, gratitude is extended to Ernst & Ernst for supporting my work with a one-year doctoral fellowship.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS--Continued

TABLE OF CONTENTS--Continued

LIST OF TABLES

LIST OF ILLUSTRATIONS

Figure Page

ABSTRACT

A methodology is proposed for aiding by computer an analysis of a plan of internal control, which is a recognized step of the audit process. The methodology consists of three major steps. First, the auditor creates a formal documentation of the company, describing those functions of the client that will be evaluated by computer. (In the case of this dissertation, the client's planned system of internal control is documented.) The documentation serves as a model, describing processes, data, people, and their interrelationships. The model is constructed in a formal language called PSL/a and stored in a computer data base.

Second, a set of rules is defined by the auditor. These rules, stated in a formal language called RULES, take the form of search operations representing the evaluation criteria that the auditor uses during the audit process. The rules describe allowable and required entities, conditions, and relations in the model of the client. The rules, therefore, describe an ideal plan of internal control and subsequently will be compared to the documented plan of internal control.

Third, the evaluation process is performed. An evaluator program reads each rule and searches through the data base under control of that rule. Any conditions in the data base that are in violation of the rule are reported as possible weaknesses in the plan of internal control. Evaluation is performed in an interactive mode, allowing the auditor to discover possible weaknesses and then exploring them further with additional rules.

The research involves elements of model formalization, data base structure, syntax formalization, and network searching. Other application areas for this research include the performance of additional audit processes, formal documentation systems, evaluation of process and information flow models.

This methodology for the evaluation of the plan of internal control has been implemented and tested on a hypothetical case study. The syntax of the modeling and rules language, the data base maintenance software, the evaluation algorithm, and the case study are detailed herein.

To indoctrinate the nonaccounting reader, an overview of auditing is presented that concentrates on the areas of special concern to this dissertation (i.e., evaluation of the plan of internal control).

A discussion of computerized audit aids shows the need for applications beyond automation of clerical tasks into the area of automated evaluation. The discussion also presents suggested techniques for quantifying audit procedures to lead toward automation. Special emphasis is given to those methodologies dealing with the system of internal control.

The proposed methodology is applied to a hypothetical test case (appearing in the appendix of this dissertation) and was not tested in a real audit situation. A discussion is presented that anticipates the acceptability and usefulness of the methodology by auditors. Interviews with auditors indicate that the automated system's acceptability should vary among auditors, depending upon their desires (or lack of desires) concerning the formalization required by the methodology and the amount of emphasis placed on internal controls by the auditor.

Extensions to the proposed methodology are suggested that enhance the type of evaluation performed, offering quantification to the evaluation process. Other suggested extensions are oriented toward making the methodology useful for audit processes beyond the evaluation of the plan of internal control.

The dissertation concludes with a discussion of the feasibility of performing an entire audit by computer. The discussion indicates those areas of auditing and computing that require further research and development.

CHAPTER 1

INTRODUCTION

A business enterprise often must describe its financial position and results of operation to people or organizations outside of the company. To add credibility to the information in the financial statements, an independent auditor is employed to attest to the fairness of the presentation. The audit process is very laborious and time-consuming. The audit team documents and studies characteristics of the operations of the client, and various tests are performed to determine the accuracy of recorded data. In the end, an opinion is written based upon the analysis of the financial information.

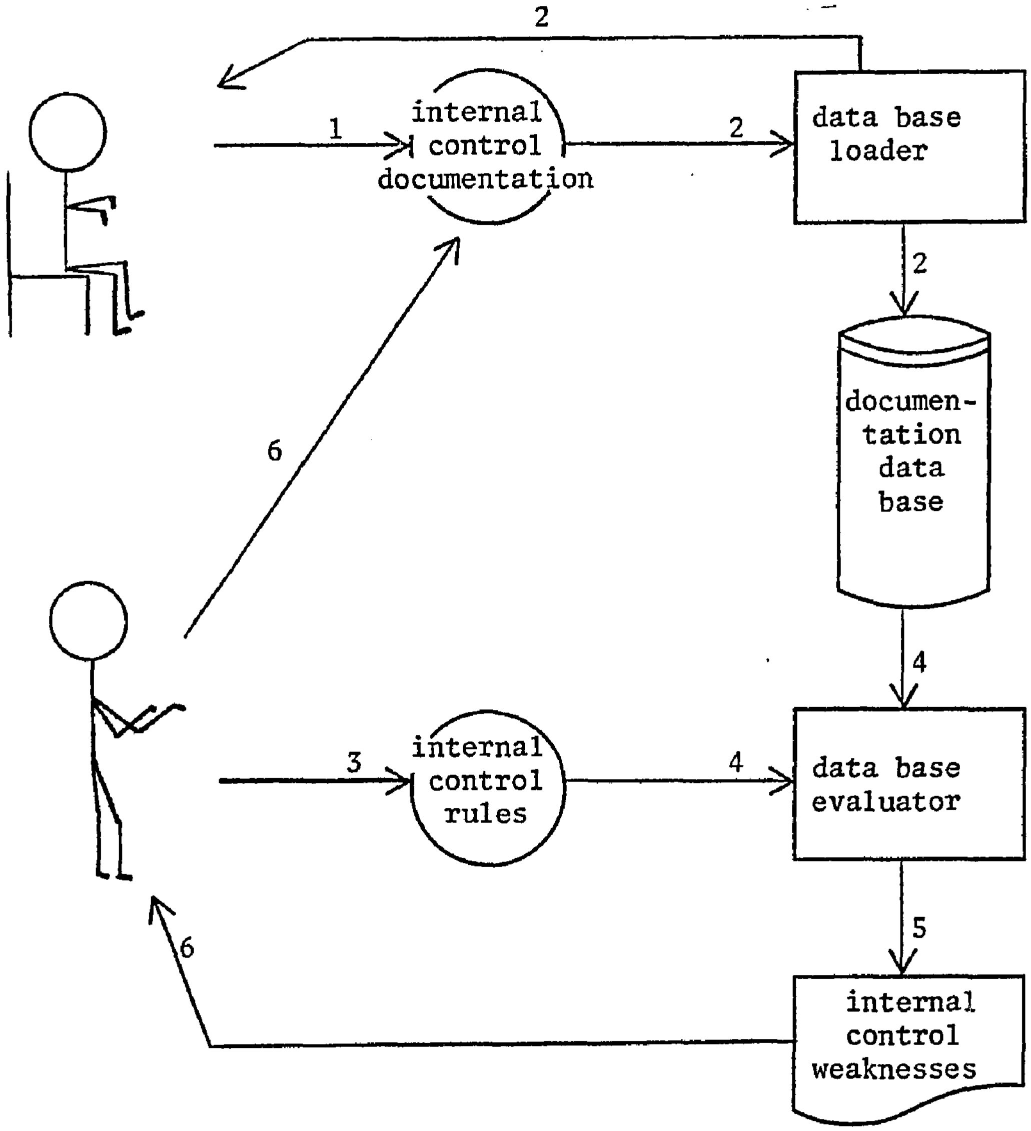
The need for computer assistance in the financial audit process is obvious. For several years, the financial auditor has made use of computer aids, but these have been clerical in nature and along the lines of data extraction and recomputation, statistical selection, generation of confirmation letters, and maintenance of audit records. Few efforts have been made to aid the audit functions of information discovery and analysis that would lead to the automation of some part of the audit process. The research presented herein has a long-range goal of automating the entire financial audit process; this dissertation proposes a methodology for one step towards that goal.

A methodology is proposed for aiding by computer an analysis of a plan of internal control, which is a recognized step of the audit process. The methodology consists of three major steps. First, the auditor creates a formal documentation of the company, describing those functions of the client that will be evaluated by computer. (In the case of this dissertation, the client's planned system of internal control is documented.) The documentation serves as a model, describing processes, data, people, and their interrelationships. The model is constructed in a formal language called PSL/a and stored in a computer data base.

Second, a set of rules is defined by the auditor. These rules, stated in a formal language called RULES, take the form of search operations representing the evaluation criteria that the auditor uses during the audit process. The rules describe allowable and required entities, conditions, and relations in the model of the client. The rules, therefore, describe an ideal plan of internal control and subsequently will be compared to the documented plan of internal control.

Third, the evaluation process is performed. An evaluator program named ICE reads each rule and searches through the data base under control of that rule. Any conditions in the data base that are in violation of the rule are reported as possible weaknesses in the system of internal control. Evaluation is performed in an interactive mode, allowing the auditor to discover possible weaknesses and then exploring them further with additional rules. The evaluation process is illustrated in Figure 1.

The auditor views the automated evaluation system as a tool to aid the audit process. The tool provides a formal manner for documenting a plan of internal control such that this documentation is created in a



1. The auditor documents the system of internal control.
2. The documentation is loaded into the computer data base and inconsistencies are corrected.
3. The auditor creates a set of rules for evaluating the system of internal control.
4. The evaluator program processes the rules against the documented system of internal controls.
5. The evaluator describes possible weaknesses in the system of internal controls.
6. The auditor studies the weaknesses and updates the documentation and applies additional rules by cycling to steps 1 and 3.

Figure 1. The Internal Control Evaluation System.

more organized manner, and it is more complete and consistent than documentation in narrative or questionnaire form. The tool also offers an automated manner for searching the documentation for information (entities and relationships) that may have been overlooked in a manual search because of the complex relationships that could exist in the documentation.

As a result of the research effort, there have been synergistic contributions beyond the development of an internal control evaluation tool. Methodologies for the documentation of information and process flow systems have been extended and generalized. A new language has been developed for specifying search patterns to be performed on an information or process flow data base, and an algorithm for executing the search has been specified. The financial audit process has been explored so as to identify areas of needed research. Above all, because of the lack of previous research in these areas, this dissertation greatly contributes to the areas of computer-aided accounting, computer-aided auditing, and automated audit evaluation performance.

This methodology for the evaluation of the plan of internal control has been implemented and tested on a hypothetical case study. The syntax of the modeling and rules language, the data base maintenance software, the evaluation algorithm, and the case study are detailed herein.

Before the methodology is described, financial auditing and "computer auditing" are overviewed to introduce the reader to these areas. The proposed methodology is described, and its probable acceptance is discussed. The dissertation concludes with suggested extensions to the research and a discussion on the feasibility of complete computerization of the audit process. A case study applying the research methodology appears in the Appendix.

CHAPTER 2

FINANCIAL AUDITING

An overview of financial auditing is presented. An understanding of the relationships among the various phases of the audit process is necessary in order to appreciate the usefulness of the research. The research methodology addresses one particular aspect of the audit process -- the evaluation of the plan of internal control.

2.1 The Audit Objective

Stockholders, bankers, creditors, and government agencies often rely on the data found in the financial statements of a business enterprise. Investors wishing to build balanced and profitable portfolios, bankers and creditors needing to measure the risk of lending money, and governmental agencies requiring proper taxation and competitive control need reliable information upon which to base decisions. These groups must put reliance on the financial statements as received, because the users of the statements were not involved in the generation of the reports.

The need for an evaluation by an external auditor is discussed by Meigs, Larsen, and Meigs (1973, pp. 2-3):

> Unaudited financial statements are not acceptable to absentee owners or other outsiders for several reasons. The statements may have been honestly but carelessly prepared. Liabilities may have been overstated as a result of arithmetical errors or through violation of generally accepted accounting principles.

> Net income may have been exaggerated because revenue expenditures were capitalized or because sales transactions were recorded in advance of delivery dates.
>
> Finally, there is the possibility that unaudited financial statements have been deliverately falsified in order to conceal theft and fraud, or as a means of inducing the reader to invest in the business or to extend credit. . . .
>
> For all these reasons (accidental errors, deviations from accounting principles, unintentional bias, and deliberate falsification), unaudited annual financial statements are not acceptable in the business community.

The objective of the financial audit is to offer assurance that the reported data are properly measured and fairly presented. The auditor attests to the fairness of the statements by appending an "audit report." It is not an objective of the financial audit to discuss weaknesses of and suggest improvements to the accounting system and organizational attributes; nor is it an objective to detect fraud. Suggested improvements and detected fraud may, however, be valuable by-products of the audit process.

The audit report is the auditor's appendix to the client's financial statements. The American Institute of Certified Public Accountants (AICPA) has carefully worded the standard short form report, and its contents provide insight into the audit process. The following report is an "unqualified (favorable) opinion" (AICPA, 1976, §509.07):

> We have examined the balance sheet of X Company as of December 31, 19XX, and the related statements of income, retained earnings and changes in financial position for the year then ended. Our examination was made in accordance with generally accepted auditing standards and, accordingly, included such tests of the accounting records and such other auditing procedures as we considered necessary in the circumstances.
>
> In our opinion, the financial statements referred to above present fairly the financial position of X Company as of

> December 31, 19XX, and the results of its operations and the changes in its financial position for the year then ended, in conformity with generally accepted accounting principles applied on a basis consistent with that of the preceding year.

The first paragraph (the "scope" paragraph) indicates to what extent the audit was performed and that generally accepted auditing standards were observed. The reader of the report places great reliance on these generally accepted auditing standards. The second paragraph (the "opinion" paragraph) presents the results of the audit. Here, it is stated that the information is presented fairly and measured properly.

Auditing standards are issued by the American Institute of Certified Public Accountants (AICPA). Accounting principles are issued by the AICPA, the Financial Accounting Standards Board (FASB), and the Securities Exchange Commission (SEC). Statements issued by these organizations are generally acknowledged by the courts. Because of the integrity of the AICPA and its members, the public may feel secure in relying on the report's summary of the audit.

2.2 Internal Auditing

As opposed to the external financial auditor's objective of attesting to the fairness of financial statements, an internal auditor's objective is to achieve the most efficient financial and managerial administration of the business. Larger corporations maintain a position for an internal auditor; in smaller companies, the internal audit task is an implicit responsibility of top management.

An internal auditor investigates and appraises the system of internal control, processing steps, flow of information and assets, and managerial control (as examples), with the goal of making these more effective and more efficient. "As a representative of top management, the internal auditor is interested in determining whether each branch or department has a clear understanding of its assignment, whether it is (effectively) staffed, maintains good records, protects ... assets properly, cooperates harmoniously with other departments ..." (Meigs et al., 1973, p. 136). The internal auditor, himself, is a part of the system of internal control.

The internal auditor's responsibilities include both accounting (those concerned with the reliability of financial records) and administrative controls (those concerned with management's decision processes leading to the authorization of transactions). Therefore, many of the procedures of the internal auditor overlap those of the external auditor. Although the techniques may be similar, the objectives are not. Because the internal and external audits overlap in many areas, techniques developed to aid one may also be useful in the other. Thus, although the methodology that is reported in this dissertation is designed as an aid to the financial auditor, it may also be used very effectively by the internal auditor, as will be further discussed in Section 8.2.2.

2.3 The Financial Audit Process

The external audit process can be segmented into four major steps. First, the auditor performs an intensive study of the client's business -- its organization, operation, and environment. The auditor studies: short-term and long-term objectives of top management; composition of the company's ownership; organization structure; types and qualities of products; flow of operations; physical facilities; types and numbers of employees; and regulatory requirements. Each company has idiosyncracies that make it different from any other business. The auditor looks for special conditions that may alter normally performed audit procedures or that may cause the need for additional procedures.

The business' environment includes the industry and the economy. The auditor studies: accounting techniques that are practiced in the particular industry, to recognize expected practices; the character of supply and demand and the competition, to anticipate some reporting desires and results; and technological developments, to assist in evaluating the handling of fixed assets. The state of the economy may control the availability of credit, affecting the soundness of the company. Compliance with regulations (e.g., wage/price controls) and laws may be verified.

Once the auditor has in mind the general aspects of the business and its environment, he or she begins to concentrate the audit process on the accounting system by studying and evaluating the system of internal control (discussed in detail in Section 2.4). This system is a set of controls implemented within the operations of the business for, among other goals, assuring proper accounting. Ideally, if the auditor found a perfect system of internal control to be in operation, there would be no need to continue the audit. However, there is no perfect system, so the auditor's evaluation of the internal control system determines the nature, extent, and timing of further audit procedures.

The third step of the audit process is substantive testing, a very time-consuming step. Here, the auditor traces transactions, recomputes balances, counts inventory, reconciles bank statements, performs analytical tests, etc., all depending on the evaluation resultant of the second step of the audit process. The objective of substantive testing is to provide direct proof of the accuracy of the reported accounting data.

The final step is the formulation of the auditor's opinion. The auditor gives one of three audit reports that are based upon the evidence obtained through substantive testing. An unqualified, qualified, or adverse opinion may be given, depending upon the nature and level of materiality of any discovered errors. (A disclaimer of opinion may be given in the case that the auditor cannot determine if the statements are reported fairly.)

2.4 The System of Internal Control

The study and evaluation of the system of internal control as part of the audit process is discussed in depth. The dissertation research is applied to a part of this step. The discussion of internal control will be limited to those aspects of concern to the external auditor.

2.4.1 The Internal Control Function

The system of internal control is comprised of those measures of the accounting system intended to promote safeguarding of the client's assets and records, assuring reliability of the client's financial data, promoting operational efficiency, and encouraging compliance with managerial policies. Assets may be stolen, misused, altered, or destroyed. These assets may be physical assets, such as machinery, inventory, and supplies, or may be nonphysical assets, such as accounts receivable and payable, documents, and records. Financial data must be properly measured, correctly recorded and used, and timely.

The second standard of fieldwork (of "generally accepted auditing standards") provides motivation for the study and evaluation of the system of internal control (AICPA, 1976, §150.02): "There is to be a proper study and evaluation of the existing internal control as a basis for reliance thereon and for the determination of the resultant extent of the tests to which auditing procedures are to be restricted." The auditor must have a thorough understanding of the system as intended and as operating. The auditor gains from the evaluation a measure of the reliance that can be placed on the system. An underlying assumption is that the better the system of internal control, the more reliable are the outputs of the system. By placing reliance on some components of the internal control system, the auditor can reduce the extent of later auditing procedures. The quality of the internal control system, then, dictates the type and amounts of testing that the auditor will have to perform.

2.4.2 Elements of Internal Control

The system of internal control can be perceived as a set of required or disallowed actions or relationships of people, processes, and physical entities of the business. Types of controls that are commonly practiced are now presented.

Transactions are defined as "exchanges of [physical or nonphysical] assets or services with parties outside the business entity and transfers or use of assets or services within it" (AICPA, 1976, §320.20). A good system of internal control assures the proper authorization, execution, and recording of all transactions. "Authorization of transactions refers to management's decision to exchange, transfer, or use assets for specified purposes under specified conditions" (AICPA, 1976, §320.21). "Execution of transactions includes the entire cycle of steps necessary to complete the exchange of assets between the parties or the transfer or use of assets within the business" (AICPA, 1976, §320.23). "Recording of transactions comprehends all records maintained with respect to the transactions and the resulting assets or services and all functions performed with respect to such records" (AICPA, 1976, §320.24).

Several controls that assure proper authorization, execution, and recording of transactions are suggested by Arens and Loebbecke (1976, pp. 161-168) as follows:

1. The responsibility of performing each duty must be assigned to specific employees. Reviews should indicate that the intended lines of authority and responsibility are actually being followed.
2. The responsibilities of execution and recording of transactions must not be performed in the same department, because lack of this separation may lead to bias in the records. Additionally, it is preferable that record keeping be performed by an independent department.

3. The responsibilities of authorization and execution of transactions that involve a common asset must be segregated. This is also true for the responsibilities of execution and recording. The separation serves a two-fold purpose: the detection of errors (accidental), and the prevention of irregularities (intentional). By combining the responsibilities of authorization and execution, the authorization function has essentially been eliminated, because self-authorization offers no controls. When the functions of execution and recording are combined, there is excessive risk of the asset being disposed for personal gain and the records being adjusted to cover the irregularity. Similarly, it is undesirable for one person to be responsible for the recording of a transaction from its inception to its final posting in the general ledger, which would allow for the undetected propagation of errors.
4. Documents and records must be designed such that adequate information is recorded to trace the flow of all information through the accounting system.
5. Physical precautions offer a measure for safeguarding assets. These precautions include storerooms, locks, fireproof vaults, safety deposit boxes, and mechanical devices that control the execution or recording of transactions.

2.4.3 The Study and Evaluation Process

The audit process concerned with internal control consists of three steps. First, a study and evaluation are made of the plan for internal control. Second, it is determined to what extent the plan has been implemented and is operating. Third, the system as implemented is studied to determine that it continuously operates as planned.

The plan of internal control is management's concept of what controls are intended to be in operation. The plan is studied by collecting descriptions of components of the plan through inspection of procedures manuals and other documents (organizational charts, chart of accounts, etc.) and interviews with management. Flowcharts and questionnaires are common aids for recording of the plan of internal control by the auditor.

To evaluate the plan of internal control, the auditor must define a set of internal control objectives (often retained mentally) that he or she expects the client to meet. The control procedures that are included in management's plan are compared to these control objectives, and a set of weaknesses and strengths in the plan of internal control is derived. The auditor evaluates the weaknesses and strengths as criteria for further audit tests. Some of these additional tests are applied in further evaluating the system of internal control, and other additional tests are designed for later substantive testing.

The next two steps are referred to as compliance testing. Once the plan has been evaluated, it must be determined to what extent the plan has been installed. This process involves the tracing of a few transactions through the system to verify that all actions that are expected to happen according to the plan of internal control actually do happen.

Finally, the effectiveness of the operational system is evaluated. Some errors are expected to remain undetected by the system; it must be determined that the occurrence rate of these errors is within tolerable limits. Further, viewing the system in action may uncover some operational difficulties or inadequacies that were not noted from the study of the plan.

The first of these three steps is the subject of this dissertation, which proposes a methodology for automation of evaluating the plan of internal control.

2.5 Audit of Computer Systems

The terms "computer auditing" and "EDP auditing" are widely used in the field of accounting in reference to the audit of financial information produced by computers. There have appeared many discussions in the literature (for example, Chandler and Mullin, 1971; Auditing Advanced EDP Systems Task Force, 1975; Horwitz, 1976; John and Nissen, 1970; Lewis, 1971; Porter, 1974; Rittersbach and Harlan, 1974; Study Group on Computer Control and Audit Guidelines, 1970, 1975) covering the concerns and difficulties involved in such audits.

The audit is greatly complicated by the introduction of the computer. Generally, the computer audit is divided into two areas of study: general controls and application controls. General controls related to EDP include: 1) the plan of organization and operation; 2) control over program design, implementation, and correction; 3) hardware controls; and 4) controls over access to hardware and data files (AICPA, 1976, §321.07).

Application controls are subdivided into input controls, processing controls, and output controls. Input controls should assure that all inputs are authorized "and that data ... have not been lost, suppressed, added, duplicated, or otherwise improperly changed" (AICPA, 1976, §321.08). Much control should be exercised over the original creation of the data. Is the preparation authorized? Is the format correct? Are the data that actually enter the computer the same authorized data which were originally created?

"Processing controls are designed to provide reasonable assurance that electronic data processing has been performed as intended for the particular application; i.e., that all transactions are processed as authorized, that no authorized transactions are omitted, and that no unauthorized transactions are added" (AICPA, 1976, §321.08). Here, the auditor must determine that the programs perform what is required and *only* what is required.

Assurance of the accuracy of outputs and assurance that outputs are received by authorized personnel are the goals of output controls (AICPA, 1976, §321.08).

This short discussion is presented only to differentiate the proposal of this dissertation from current "computer audit" practices. The computer audit, as discussed above, involves the audit *of* computer systems by manual or automated means. This dissertation, however, involves the audit *by* computer of a manual or automated accounting system. The distinction must be emphasized because the two areas are not related but are often confused.

CHAPTER 3

COMPUTERIZED AUDIT AIDS

This chapter provides the motivation for the research effort. Presented first is a discussion of the need and desire to automate portions of the audit process. This is followed by presentations of approaches taken by other researchers. Finally, an overview of the proposed methodology is presented.

3.1 Motivation for Developing Automated Evaluation Aids

This research effort stems from two avenues. The first is based on the need for methods to aid the audit process so as to reduce audit cost, time, and error. The second motivation is from a pure research desire to determine how much of the audit process (and other evaluation processes) can be performed by computer. Both of these areas are now introduced.

3.1.1 The Need for Aids to the Audit Process

The audit process can be viewed as processes of collecting, classifying, recording, summarizing, and evaluating accounting and procedural information.

Collecting, recording, and classifying data for audit evaluation are made very difficult by the volume and diversity of data that are involved. Documentation of the system of internal control, for example, has in the past taken the forms of questionnaire answers, narrative descriptions, and flowcharts. The use of questionnaires may limit the extent of the documentation in that the questionnaires do not encourage the discovery of information outside the realm of the questions. In the case of narrative descriptions, the completeness and comprehensibility will depend upon the documentor's interpretation of his or her own perception of the system and upon the documentor's writing style. The use of flowcharts may involve inconsistencies due to personal preferences of the documentor, and the use may involve incompletenesses because of the tendency to omit information that appears to be obvious to the documentor at the time the description is recorded. Classification of the data is necessary in order to make summarization and evaluation possible. Informal methods of classifying and recording are open to inconsistencies, sometimes resulting in inconsistent evaluations.

A method of documentation more formal than questionnaires, narratives, and flowcharts that allow too many stylistic freedoms is necessary to encourage complete recording, consistent identification, objective interpretation, and proper evaluation of data (e.g., the description of the system of internal control). A formal documentation scheme allows for a formal retrieval scheme; that is, evaluation of the audit information is aided by making the desired information easier to access. The first portion of the proposed methodology of this dissertation is a formal documentation language and a computer system for recording, maintaining, and extracting a set of documentation.

3.1.2 The Research Motive for Automated Evaluation

There is much to be gained beyond simplifying the auditor's tasks in the effort to automate portions of the audit process. The study of audit functions from a new perspective (i.e., attempting to automate them) allows underlying audit principles and objectives to be uncovered and questioned. The answers may involve new approaches to auditing and perhaps a more refined definition of audit objectives. Areas that currently involve subjective analysis may be made quantifiable and thus allow for objective analysis. This research effort delves into the audit process to reveal new areas for investigation.

3.2 Computer Aids for the Audit Process

A long-range goal of computer auditing may be moving toward the complete automation of the audit process. It is realized that this goal is currently considered unachievable, but it offers a direction in which to guide the research efforts. The levels of involvement of the computer in the audit process are illustrated in Figure 2. This section discusses research that has been conducted and techniques that have been implemented dealing specifically with the use of the computer to aid the audit process.

The discussion begins at the lower end of Figure 2's continuum with what are referred to as general audit aids, since their nature is to replace much of the manual and clerical chores of the auditor, but not actually to perform a recognized step of the audit process.

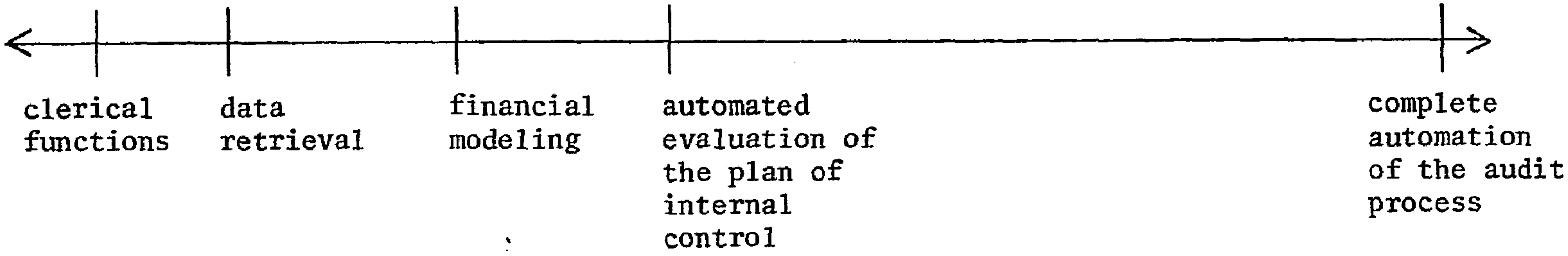

clerical functions
data retrieval
financial modeling
automated evaluation of the plan of internal control
complete automation of the audit process

Figure 2. Levels of Computer Involvement in Aiding the Audit Process.

3.2.1 Clerical Functions

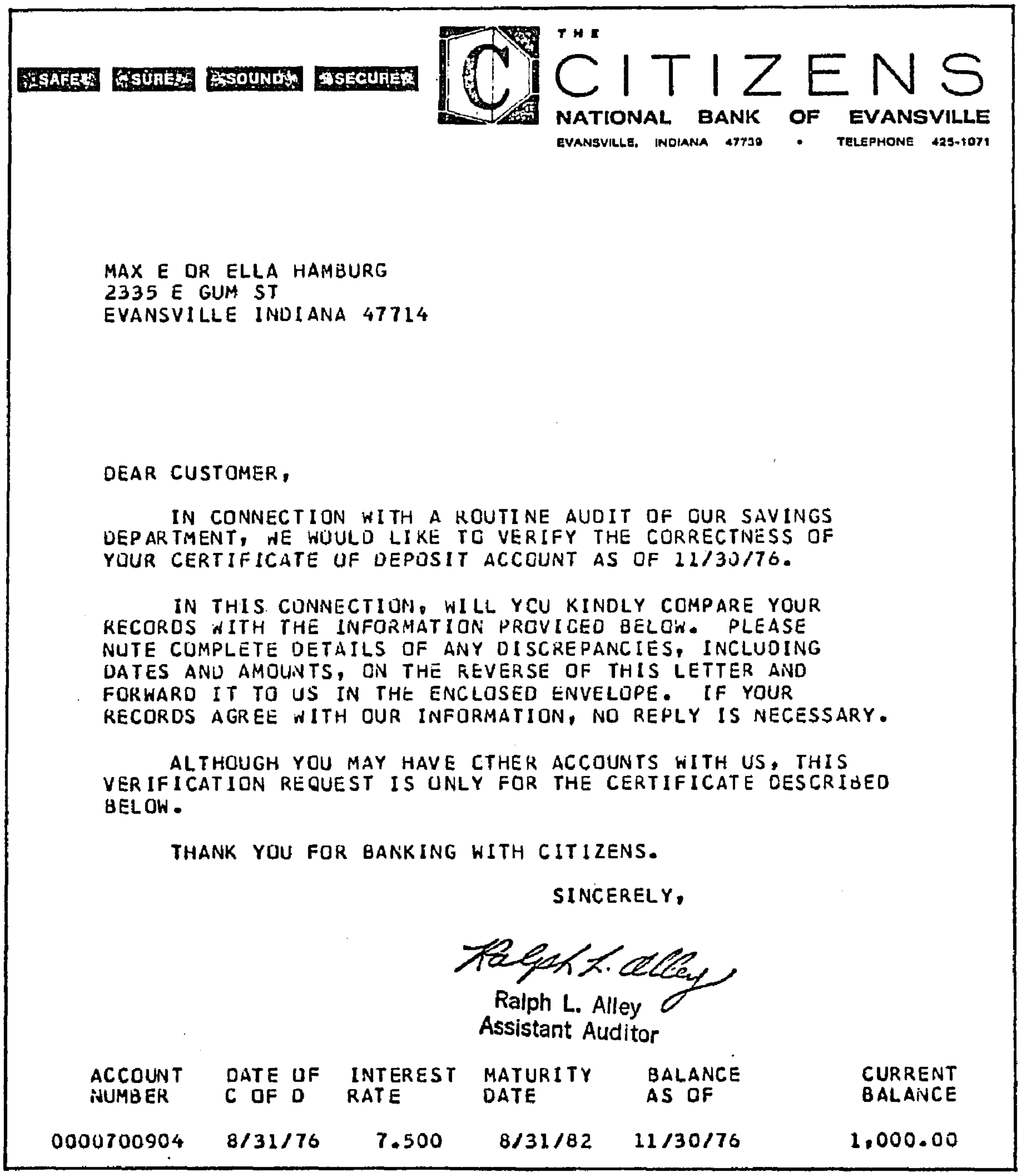
There is much record-keeping and paper work involved in the audit. Data are collected from many sources for analysis. These data must be maintained in an organized manner, and computer files offer good storage media. Analysis of the data involves computing sums, searching for exceptions, stratifying values, and comparing files. A computer can perform these tasks faster and more reliably than they can be performed manually. Many of such clerical tasks can be handled by a computer.

A widely used application is the generation of confirmation letters and the maintenance of related information. A confirmation letter (such as appears in Figure 3) is sent by the auditor to a customer of the client. The letter asks the customer to verify the balance of his or her account payable to the client. The letters are sent to a statistical (perhaps stratified) sampling of customers. The computer can aid the process by evaluating the data to aid in the determination of the sampling technique to be used, performing the random sampling, generating the confirmation letters, and maintaining and analyzing the responses to the letters.

3.2.2 Retrieval of Accounting Information

If the client performs accounting functions by computer, then much of the information needed for the audit is stored on machine-readable media. The auditor needs to access and test these data. Programs have been developed that offer the retrieval capability.

"Generalized audit packages" (GAP) (Adams and Mullarkey, 1972, pp. 39-66) are programs primarily used for accessing and evaluating data



SAFE SURE SOUND SECURE

THE CITIZENS NATIONAL BANK OF EVANSVILLE
EVANSVILLE, INDIANA 47739 • TELEPHONE 425-1071

MAX E OR ELLA HAMBURG
2335 E GUM ST
EVANSVILLE INDIANA 47714

DEAR CUSTOMER,

IN CONNECTION WITH A ROUTINE AUDIT OF OUR SAVINGS DEPARTMENT, WE WOULD LIKE TC VERIFY THE CORRECTNESS OF YOUR CERTIFICATE OF DEPOSIT ACCOUNT AS OF 11/30/76.

IN THIS CONNECTION, WILL YCU KINDLY COMPARE YOUR RECORDS WITH THE INFORMATION PROVICED BELOW. PLEASE NUTE COMPLETE DETAILS OF ANY DISCREPANCIES, INCLUDING DATES AND AMOUNTS, ON THE REVERSE OF THIS LETTER AND FORWARD IT TO US IN THE ENCLOSED ENVELOPE. IF YOUR RECORDS AGREE WITH OUR INFORMATION, NO REPLY IS NECESSARY.

ALTHOUGH YOU MAY HAVE CTHER ACCOUNTS WITH US, THIS VERIFICATION REQUEST IS ONLY FOR THE CERTIFICATE DESCRIBED BELOW.

THANK YOU FOR BANKING WITH CITIZENS.

SINCERELY,

Ralph L. Alley

Ralph L. Alley
Assistant Auditor

ACCOUNT NUMBER	DATE OF C OF D	INTEREST RATE	MATURITY DATE	BALANCE AS OF	CURRENT BALANCE
0000700904	8/31/76	7.500	8/31/82	11/30/76	1,000.00

Figure 3. Confirmation Letter.

stored on magnetic tapes. The files under evaluation are usually "history" files such as master files, balance forwards, and account balances (e.g., accounts receivable and payable).

To use a GAP, the auditor fills out a set of forms that are then interpreted by the GAP. The forms fall into three categories. First, the general system is specified. The input/output media and devices are identified and run parameters (error limits, dates, random number generator seeds, etc.) are declared. Second, all files (input, output, and work files) are specified. The file organizations are declared and data fields are named and described. Third, the processes to be performed on the data are specified. Fields to be selected, merged, added, compared, stratified, and printed are identified.

Execution of the GAP against the data files is generally done by one of two methods. One is a two-step process: first, a program is generated by automatically selecting program modules and linking them together; second, the generated program is executed, performing the desired extraction functions. The other method does not involve the generation of an intermediate program, but rather calls for execution by the entire GAP. Most GAPs perform all desired processes on the data with one pass of the data files.

GAPs can be used on many different computers, can be used with many standard file structures (i.e., sequential, direct, and indexed sequential), and the language of the client's program is irrelevant (since the GAP processes data files and does not come in contact with the program that inputed or generated the data files).

Along the lines of GAPs, Will (1976) has designed an interactive data extraction and evaluation tool called ACL (Audit Command Language). The user may define: 1) a file structure and its data constructs; 2) temporary variables and formulae; 3) report formats; 4) logical tests to be performed on the data; and 5) a series of commands in the form of a "job." The following commands may be applied to the data: count, extract, list, select, sample, sort, total, mean, standard deviation, variance, minimum, range, kurtosis, skewness, correlate, histograph, regress, evaluate, summarize, and verify. These commands are executed interactively against an online data bank. In such a manner, the auditor may perform computations, evaluate the results, and perform further studies based on the findings.

The design of the system (Will, 1975) includes a data bank and a model bank. The data bank contains all the accounting data upon which tests are to be performed. The model bank consists of application programs and testing packages. These banks are interfaced in such a way that the application programs are independent of the data bank's structure. The user interacts with the interface system to execute the programs against the data.

3.2.3 Discussion of General Computer Aids

The aids discussed in Section 3.2 were developed to replicate manual processes on a computer. The intent of their design is to aid the auditor in performing his or her tasks faster, as opposed to replacing audit tasks. The contribution in these methods toward the automation of additional audit functions is minimal. Their presentation is included to contrast this class of work with the types presented in Section 3.3 and with the proposed dissertation methodology.

3.3 Techniques for Evaluating Internal Control Systems

The contribution of this dissertation is a method by which an evaluation of the plan of internal control can be aided by computer. The research may be extended later so as to encompass additional audit functions to be automated. Therefore, methods developed by other researchers that aid the audit process through the use of a computer are investigated. Although the objective of these researchers was not to automate the audit function, their developments are of interest as methodologies that could be applied in the study of automated auditing.

The implications and applications of the presented methods on the dissertation research are discussed at the conclusion of this section.

3.3.1 Analyzing Adequacy of Controls through Flowchart Analysis

It has been suggested by numerous people (as examples: Committee on Auditing Procedure, 1949; Anderson, 1972, pp. 36-54; Stickler, 1968, pp. 412-415) that the system of internal control (i.e., the flow of those processes and information that are of concern in the audit) be flowcharted. The auditor may then evaluate the system of internal control from the flowchart. This technique is often referred to as "analytical auditing."

Flowcharting offers the auditor a manner in which he or she may pictorally describe on paper the flow of the system. Such a picture allows the flow of processes and information to be readily observed, and all processes and information that directly interact with a particular process or a set of data may also be readily discovered. Because of the visibility of the entire system, it is easier to determine that the description is complete and correct.

The alternative to the flowchart is a written description. Due to the nature of writing (a linear presentation), aspects of the system that are related might not be described near one another, and the flow may not be very apparent. A narrative description is prone to incompleteness and inconsistencies, and its comprehension is difficult (refer to Section 3.1.1).

Important to the comprehension of the flowchart is the locality of the symbols on the chart. The chart is divided into columns, each representing a department or responsibility area of the client. A process is drawn in the column of the department in which it is performed, and the processing symbols are linked together. Therefore, flow across the form represents the flow among departments, while flow down the form represents the flow over time.

This flowchart format was used in the development of the case study. It may be referenced in the Appendix.

3.3.2 Computer-Based Documentation of the System of Internal Control

Aids in computer systems analysis and automation of computer systems design are objectives of the ISDOS project (Teichroew and Sayani, 1974). A major result of the overall project is the PSL/PSA system.

PSL, Problem Statement Language, is a language for describing a desired system. PSA, Problem Statement Analyzer, is a software system for storing and maintaining the description in a data base and handling the extraction of information from the description for analysis. The description is a static representation of a real-world system serving as a very formalized set of documentation of a system design, so that evaluators (whether they be software or human) may access any portions of the description in an organized manner. The analysis involves heuristic procedures of studying the existence (or absence) of entities, and how they are interrelated.

Rather than limiting the use of PSL to describing information systems, Lieberman, Nunamaker, and Warren (1975) suggest the use of PSL for describing accounting systems (whether they be manual or automated accounting systems). Refer to Figure 4 for a PSL example. The following advantages are gained by describing the accounting system in a formal language: 1) the documentation is made available for computer maintenance and processing; 2) the syntax and semantics of the information are independent of who enters the information; 3) determination of completeness and consistency of the documentation may be automated; 4) orderliness of the set of documentation is independent of order or timing of entry of information; and 5) updating of the documentation is simplified.

The concept of PSL documentation is similar to that of flowcharting (although different in appearance). Both methods involve the recording of objects, activities, and people, and how they interact.

```
REAL-WORLD-ENTITY general manager;
   RESPONSIBLE-MANAGER is Frank Obrien;
   SUBPARTS ARE marketing-department, manufacturing department,
      operations-department, and financial department;
   DESCRIPTION;
      The general manager is responsible for overseeing the four major
      departments;
   SUBPARTS ARE transportation-department, production-department, and
      purchasing-department;

RESPONSIBLE-MANAGER R. Smith;
   RESPONSIBLE FOR manufacturing-department;

PROCESS parts-requisitioning;
   RECEIVES parts-requisition;
   GENERATES picking-ticket, parts-backorder;
   AUTHORIZATION IS INITIALS ON parts-requisition;
   AUTHORIZED BY MANAGER OF prod-department;
   PROCEDURE;
      When the production department needs parts to be used in the
      assembly of the product it sends a Parts Requisition to the Data
      Processing Center.  If enough parts are on hand to fill the
      request, a picking ticket will be sent to the warehouse indi-
      cating how many of each part to send to the Prod. Dept.  If an
      insufficient amount of parts is available to fill the order, as
      many parts as available will be sent and a parts backorder will
      reflect the difference;

CONDITION parts-level-low;
   TRUE WHILE;
      Any requisition lowers the quantity of a part on hand to below
      the reorder point;

EVENT purchase-flagging;
   TRIGGERS purchase-request-process;
   WHEN parts-level-low BECOMES TRUE;

INPUT parts-requisition;
   CONSISTS OF part-description, part-quantity-requisitioned;

GROUP part-description;
   CONSISTS OF part-number, part-name;

ELEMENT part-quantity-requisitioned, part-number, part-name;
```

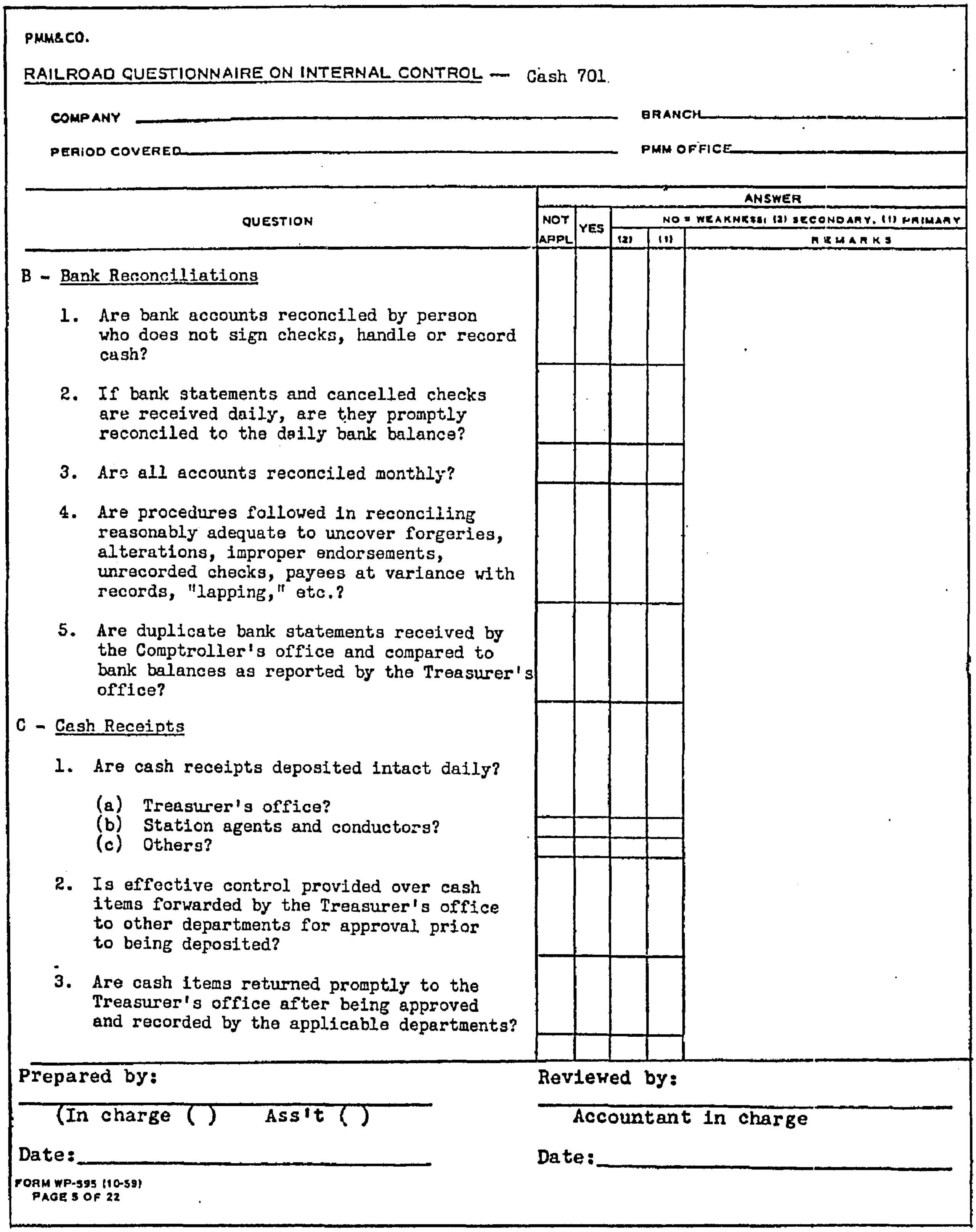
Figure 4. PSL Description of an Accounting System.

PSL may be thought of as an automated form of the flowchart technique, and the advantages of PSL over flowcharting are included in the above list.

While PSL may be used for describing the accounting system, it may also be used for describing the system of internal control. The extant PSL does not include the necessary terminology for documenting accounting systems; the language content, although not its structure, needs to be extended. The original reference (Lieberman et al., 1975, pp. 1284-1285) suggests some of these extensions as examples. Kehl (1977) has extended PSL with specific "name types" so as to result in a more suitable language for modeling an accounting system and system of internal control. His approach has been to attempt to enumerate those inputs, outputs, processes, and relationships that may be desired for documenting an accounting system.

3.3.3 Questionnaires for Documenting and Evaluating Internal Control

The use of internal control questionnaires is common practice among auditors. Questionnaires are individualized for the particular industry under audit and for each particular area of the accounting system. They are normally comprised of a series of questions in which "yes" or "adequate" are desired answers. (An example of a questionnaire is presented as Figure 5.) For any "no" or "inadequate" answers, the auditors writes a comment next to the question clarifying the situation or calling attention to the weakness in the system. Use of a questionnaire offers the following advantages: 1) questions are prepared in



PMM&CO.

RAILROAD QUESTIONNAIRE ON INTERNAL CONTROL — Cash 701

COMPANY ______________________ BRANCH ______________________

PERIOD COVERED ______________________ PMM OFFICE ______________________

QUESTION	ANSWER				
	NOT APPL	YES	NO = WEAKNESS: (2) SECONDARY, (1) PRIMARY		
			(2)	(1)	REMARKS
B - Bank Reconciliations					
1. Are bank accounts reconciled by person who does not sign checks, handle or record cash?					
2. If bank statements and cancelled checks are received daily, are they promptly reconciled to the daily bank balance?					
3. Are all accounts reconciled monthly?					
4. Are procedures followed in reconciling reasonably adequate to uncover forgeries, alterations, improper endorsements, unrecorded checks, payees at variance with records, "lapping," etc.?					
5. Are duplicate bank statements received by the Comptroller's office and compared to bank balances as reported by the Treasurer's office?					
C - Cash Receipts					
1. Are cash receipts deposited intact daily?					
(a) Treasurer's office?					
(b) Station agents and conductors?					
(c) Others?					
2. Is effective control provided over cash items forwarded by the Treasurer's office to other departments for approval prior to being deposited?					
3. Are cash items returned promptly to the Treasurer's office after being approved and recorded by the applicable departments?					

Prepared by: ______________________
(In charge ( ) Ass't ( )

Date: ______________________

Reviewed by: ______________________
Accountant in charge

Date: ______________________

FORM WP-595 (10-59)
PAGE 5 OF 22

Figure 5. Internal Control Questionnaire.

advance, eliminating wasted time during field work; 2) a standard set of questions allows for a consistent tool of measurement for the auditor; and 3) a standard set of questions assures that the auditor will not overlook any of the areas handled by the questionnaire (this may also be a disadvantage in that the auditor may feel constrained by the questionnaire and might not venture into omitted questions concerning special cases).

Brown (1962, pp. 50-56), although in favor of the questionnaire approach, recognizes a particular problem in its practice. "Yes/no" and "adequate/inadequate" answers involve subjective appraisals, and several auditors may evaluate the effectiveness of an internal control differently from one another. The nature of questionnaires tends to disguise the importance of their purpose, and auditors may fail to give them their required consideration beyond the "yes" or "no." Further, after the individual questions have been considered and answered, the technique offers no method for forming a judgment on the overall effectiveness of the system of internal control.

Brown (1962) proposes that a questionnaire describing a perfect system of internal control be designed, and that weights be assigned to the questions. A "yes" answer would be replaced by the weighted value, and a "no" answer would be replaced by a zero value. The overall effectiveness of the system, then is the ratio of the attributable values to the total possible weight. This ratio (so claims Brown, 1962, p. 53) can be used "rationally and precisely to determine the extent of [compliance and substantive] testing required."

3.3.4 Determining Proper Levels of Compliance Tolerance through Simulation

Burns and Loebbecke (1975, pp. 60-70) recognize a problem in determining tolerable compliance levels for compliance testing (the second step of the analysis of the internal control system). In practice, conservative compliance levels are set without any empirical or theoretical support. Their arbitrariness may lead to wasted audit procedures (and expenses) or may lead to insufficient audit evidence.

A suggested solution technique to selecting an appropriate compliance level is the use of simulation. A computer program models a particular internal control system (such as the raw materials inventory accounting system). Two important areas are of special concern in the model. First, each type of error that could occur in the system must be identified and described quantitatively. In the program, the occurrences of errors will happen at random intervals according to a prescribed frequency. Second, all internal controls that detect or prevent errors must be identified. The probability of any control failing is an exogeneous variable, which is defined at the time of the simulation run.

The simulation is run several times, each run using a different prescribed tolerance level (the complement of the probability that internal controls will fail). Resultants of the simulations are expected dollar values of errors in the system. The materiality of each of the error amounts is weighed by the auditor (a subjective evaluation), and, in so doing, the auditor selects the most desirable level of compliance to be applied during actual compliance testing.

3.3.5 Evaluation of Internal Control through Stochastic Modeling

The system of internal control may be viewed as a series of processes and decisions. At each step of the system, there exists some probability that an error will occur. The errors propagate through the system, affecting the reliability of the final outputs of the system. Yu and Neter (1973, pp. 273-295) recognize this phenomenon and propose a stochastic model for tracing the flow of error propagation.

The types of functions performed within any system may be dichotomized as transformation operations and decision operations. A transformation operation is a process that receives an input and, in accordance with prescribed rules, generates an output. For each input, a probability is assigned that if the input is of a particular error state then the output is of another error state. A decision operation distributes inputs between two output classifications. For example, the verification of a payroll time card may lead to time card processing or back to time card preparation, depending upon whether an error is discovered on the time card. The distribution process is also prone to error, and values are assigned expressing the probability of the occurrence of error.

Given an input vector stating the probability of the different error states for the original input into the system, a vector reflecting probabilities of error in the output can be generated. This output vector is then used as the input vector for the next process in the system flow, yielding another output vector. This is carried through the entire system until the final output is reached. The final output vector represents the reliability of the results of the accounting system.

In operation, the initial input vector, the transformation probability values, and the decision probability values must be specified. The probability measures may be obtained through the evaluation of sampling studies.

There are many applications for the stochastic model. Its primary purpose, as already discussed, is to form a measure of reliability of the final outputs (account balances) of the accounting system. Also, weak points in the system's flow may be observed by graphing the process steps against their corresponding error probability. The impact of unusual error occurrences may be studied by altering an input vector or probability matrix at a particular point in the flow. As the system changes over time (through hiring of new personnel or automating procedures, as examples), the probability matrices will change, and the effect may be studied.

Other research concerning stochastic models is worth noting at this point. Cushing (1974, pp. 24-41) proposes a model that incorporates the following probabilities for each process: 1) that the process is correctly executed prior to administering the control procedure; 2) that the control step will detect and signal an error given that one exists; 3) that the control step will not signal an error given that none exists; 4) that the correction step will correct an error given that one exists and has been signaled; and 5) that a failure of the control step will be detected and no correction made given that the control signals an error when none exists. These probabilities are combined to indicate a reliability measure of the outputs of the system. A process may involve many controls and many error types. A cost function is defined as consisting of the cost of performing the control procedure, the average cost of correcting the error, and the average cost of an uncorrected error. This allows for cost-benefit analysis of implementing a control procedure. Bodnar (1975, pp. 747-757) discusses the feasibility of implementing such an approach.

Hannum's model (1974, pp. 311-322) recognizes the number of time units that elapse between processes. The contributions per time unit of correctly versus incorrectly operating processes are accounted for, along with costs of reporting and corrective actions, and probabilities of errors occurring or not being detected. Functions of the variables express long-run expected contributions for given reporting schedules (Hannum's "reports" correspond to Yu and Neter's, 1973, "transformation outputs").

3.3.6 Compliance and Substantive Testing through Statistical Sampling

The goal of compliance and substantive testing is to determine appropriate values of account balances at an acceptable level of confidence. Practiced auditing searches for evidential matter on which to base the assurance of an account's balance. How much evidence is required? The auditor could compile all transactions that have affected a particular account over the past year. For many accounts, this process, obviously, would be too time-consuming and costly to be feasible.

Statistical sampling is a tool with mathematical basis that allows the auditor to infer, from observation of a portion of a population, certain attributes of the entire population. The population may be the transactions affecting an account, a set of physical entities, or the collection of customer accounts, as examples. Statistical sampling is not a perfect science, and there is always the possibility that the sample is not representative of the population; this difference is referred to as "sample error."

To select a proper sample size, the auditor specifies three variables, precision, confidence level, and an expected attribute rate or value. Precision is the acceptable range around an expected value of the occurrence of some event. For example, an error may be expected to occur in 5.3% of the recorded transactions. The auditor may accept a precision of 2%, or an expected error rate between 3.3% and 7.3%. Confidence level is a measure of how much the auditor needs to be assured that the sample represents the entire population. For example, a 95% confidence level means that the auditor can expect that 95% of the time the results based on the chosen sample will represent the same characteristics (within the specified precision range) of the entire population. Few guidelines are given to the auditor for selecting precision and confidence levels, although the technique is practiced and approved by the AICPA. Research described in Section 3.3.4 is concerned with aiding the determination of these variables.

Equipped with the size of the population, an expected occurrence rate or value, a precision level and level of confidence, the auditor refers to a table to determine the proper sample size. The sample is then selected and evaluated as if it were the total population. If a higher-than-expected error rate is found, or if other anomalies are discovered, then further audit tests would be applied aside from statistical sampling.

3.3.7 Discussion of Internal Control Evaluation Aids

The techniques discussed in Section 3.3 were designed with the intent of aiding or completely performing a particular audit function. They are of a nature such that they can be extended, considered separately or in conjunction with one another, to perform additional audit functions. Because they can be considered as approaches toward automation of the audit process, they are of much importance to this dissertation.

The proposed methodology of this dissertation evolves from the conjunction of three of the aforementioned techniques or systems (flowcharting, PSL, questionnaires). Flowcharting was borrowed as a technique by which an auditor can record the plan of internal control in a manner which he or she can more easily understand. If the internal controls are to be evaluated, then a description of the controls must be provided. Flowcharting provides a good medium for that description (or "documentation"), because it is unambiguous and simpler to perceive than a narrative description.

PSL was borrowed as a technique for formalizing the flowchart description of internal control so as to make it computer-accessible and processable. PSL has been extended (first by Kehl, 1977, and again independently of Kehl in this dissertation, Chapter 4) to better fit the audit environment. By storing the PSL documentation of the plan of internal control in a computer data base, consideration may be given to computerizing an evaluation of that plan of internal control. (This is the subject of Chapter 4.)

Finally, the questionnaire technique was borrowed as an approach to performing the evaluation method in this dissertation. A language was developed (as a contribution of this dissertation) for expressing the evaluation questions. Although the language does not appear in the same form as that of the questionnaire entries, the concept of determining if certain controls are met by the system (resulting in "yes" or "no" responses) is retained. (This is the subject of Chapter 5.)

The proposed methodology applies the questions of the questionnaire (in a different format) to the plan of internal control (as represented by the PSL documentation stored in the data base, which in turn is based upon the flowchart description). The result of processing is the identification of potential weaknesses in the system of internal control. (This is the subject of Chapter 6.)

The other techniques of Section 3.3 are important in the consideration of extensions beyond the scope of this dissertation. The weighting of the questions, as suggested by Brown (1962), is the next natural extension to the methodology. Although compliance testing is not a concern of the current application of the dissertation, it is the next logical application area, and providing objective measurement methods such as that suggested by Burns and Loebbecke (1975) should be considered. Evaluation of the interaction of internal controls will be a necessary addition to the methodology of automating the audit process, and stochastic modeling is one approach to this area. The application to compliance and substantive testing will necessarily involve statistical sampling as the automation research encompasses more of the audit process. These ideas are further explored in Chapter 8.

3.4 Overview of the Proposed Methodology for Automated Evaluation

The research effort has been applied to one step of the audit process, the evaluation of the plan of internal control. A starting point was needed that did not encompass more than could be handled by a dissertation effort. This starting point was chosen because it is the first major step of the audit process, and it fits very well into the proposed methodology.

In the nonautomated evaluation of the system of internal control of a company, the auditor begins by interviewing management, studying documents and observing processes to determine the intended and operational controls. The auditor should have an ideal system in mind to aid his or her awareness of any incompletenesses that may exist in the collected information. Once the system has been identified, it must be evaluated. An auditor generally has learned the constructs of a "good" system of internal control through training and experience. This knowledge and the aid of written guidelines are used by the auditor to identify strengths and weaknesses in the system. Since no two companies are identical, the auditor must be aware of the special circumstances and characteristics of a company that may require additional controls or cause exception to standard controls. The overall evaluation is based on the discovered strengths and weaknesses and their possible effects on the outputs of the accounting system.

If a computer is to perform an evaluation of the plan of internal control, then the computer must have available to it what the auditor has available to him or her: the description of the internal control system and the constructs of a "good" system to serve as a basis for comparison. These two information sets comprise two of the three basic functions of the methodology proposed here. The third function is evaluation processing.

An overview of the proposed methodology for the automated evaluation of the plan of internal control is now presented. First, a formal description of the client's plan of internal control is defined. This documentation is written in a language called PSL/a (Problem Statement Language/accounting) that allows for the description of entities and activities and their interrelationships. A loader program stores the documentation into a computer data base. A set of programs aids in the maintenance of the data base by adding, deleting, correcting, and reporting the contents of the documentation.

Second, a set of rules is specified that corresponds to the auditor's evaluation criteria (i.e., to the internal control questionnaire, perhaps). These rules describe required and illegal entities, activities, and relations of an acceptable system of internal control, and subsequently will be compared to the documentation resulting from step 1. The rules are written in a formal language called RULES and maintained as a simple sequential file.

Third, the evaluation of the plan of internal control is performed. The rules represent a desirable plan of internal control; the documentation represents what is to be evaluated; the evaluator software inputs the rules and applies them against the documented system. While processing a rule, a report is printed that describes any conditions in the documentation data base that are in violation of the rules. In use, the auditor processes the rules interactively, and upon the detection of possible weaknesses in the system of internal control, he or she processes additional rules to further explore the internal controls.

Figure 1 illustrates the flow of the methodology for internal control evaluation. The next three chapters detail these three major steps. The final two chapters discuss the implications of this methodology for the auditor and extensions of the methodology.

CHAPTER 4

FORMALIZATION OF THE PLAN OF INTERNAL CONTROL

The first step of the automated evaluation methodology is the documenting of the plan of internal control. Expressing the plan via a formal language description makes the plan of internal control computer accessible and evaluatable. The form and use of the language are described in this chapter.

4.1 Description of the PSL/a Language

To enable a computer to evaluate a plan of internal control, the computer must have access to a description of that plan. This first hurdle is satisfied by formally modeling a computer-accessible description of the client's plan of internal control. The model is described in a formal language called PSL/a (Problem Statement Language/accounting), and it is maintained in an online data base.

4.1.1 Nature of the Model.

The description of the plan of internal control will be referred to as a "model." This model is static in nature, describing the expected flows of processes and information through the company at a fixed point in time. Its purpose is to act as a formal set of documentation, enabling an evaluator to readily access any portion of the model. The language used for describing the model is based on the constructs and form of PSL, which was introduced in Section 3.3.2. PSL has been extended for this dissertation to allow for the modeling of an internal control system; this extension is called PSL/a (Problem Statement Language/accounting).

A model written in PSL/a consists of three types of components -- objects, attributes, and relations. An "object" is any entity or activity of the company. Positions, reports, files, and departments are examples of objects, as are the opening of mail, performing of cross-footing, and manufacturing of a product. An "attribute" is a description, categorization, or qualification of an object. Time of occurrence, cause of an action, and type of account are examples of attributes. A "relation" is a definition of how two objects are connected or related. If "person A owns a car," then "owns" is the relation between the objects "person A" and "car." A hypothesis of this research that has been tested only empirically here is that an adequate model for performing the methodology can be built with these three constructs.

A difficulty often arises when trying to distinguish between a relation and an activity (which is a type of object). In the case where "person A owns a car," "owns" is clearly a relation. In the case where "person A buys a car," it is not clear whether "buys" is a relation or an activity. It may be declared that if any action is involved, then it is an object, and relations may not involve action. However, this is not the intended meaning of "relation." A relation is a standardized and often-used connector between objects. If a specific activity is involved and occurs but once in the model, then it is an object instead of a relation.

Kehl (1977) attempts to specify the necessary extensions to PSL (additional objects, attributes, and relations) to enable the modeling of an accounting system or a system of internal control. However, a language that is complete cannot be expected, because it cannot be determined which objects, attributes, and relations all modelers may want to describe. With this in mind, this dissertation does not specify an absolute set of elements for the language. Instead, the language was designed from the onset to be extendable; that is, new objects, attributes, and relations may be added as necessary. In doing so, a table is provided that specifies the objects, attributes, and relations currently being used by the modeler. All software is table-driven; the software refer to this table when loading, maintaining, or accessing the model in the data base. As opposed to Kehl's approach, in which specific object, attribute, and relation types are specified, the proposed PSL/a language does not specify any types, but instead allows the user to define them as their need is determined.

The types of objects, attributes, and relations that currently are being used in this research, and that are suggested as a beginning set, are described in Tables 1, 2, and 3. The objects, attributes, and relations are more clearly illustrated in Figure 6. Although a set of object-types, attribute-types, and relation-types is suggested, this set has been proposed only as a starting point for the research. Over time, a necessary and near-sufficient set of object-types, attribute-types, and relation-types may be accumulated.

Table 1. PSL/a Object-Types.

Object-Type	Description
POSITION	the title of an employee; for example, president, mail room clerk, chief accountant
ACTIVITY	a process or transaction; for example, customer billing, verification, update of perpetual inventory
DEPARTMENT	a department name or functional area of the company; for example, mail room, receiving, accounts receivable
ACCOUNT	an account title; for example, cash on hand, accounts payable, retained earnings
REPORT	any record of data, other than an account; included are written reports, documents and computer files

Table 2. PSL/a Attribute-Types.

Object-Type	Attribute-Type	Legal Values
POSITION	PERSON	name of the person holding the position
ACTIVITY	STIMULATED	INTERNALLY, EXTERNALLY
	PROCESS	description of the activity
	PERIOD	ANNUALLY, MONTHLY, WEEKLY, etc.
DEPARTMENT	(none)	
ACCOUNT	ACCOUNT-TYPE	ASSET, LIABILITY, EQUITY, etc.
REPORT	CONTENTS	description of the report
	INITIATED	INTERNALLY, EXTERNALLY

Table 3. PSL/a Relation-Types.

Object-Type	Relation	Related to Object-Type
POSITION	MANAGES	POSITION
	MANAGED-BY	POSITION
	AUTHORIZES	ACTIVITY
	EXECUTES	ACTIVITY
	RECORDS	ACTIVITY
	EMPLOYED-BY	DEPARTMENT
ACTIVITY	AUTHORIZED-BY	POSITION
	EXECUTED-BY	POSITION
	RECORDED-BY	POSITION
	TRIGGERS	ACTIVITY
	TRIGGERED-BY	ACTIVITY
	INCOMPATIBLE-WITH	ACTIVITY
	PERFORMED-BY	DEPARTMENT
	DEBITS	ACCOUNT
	CREDITS	ACCOUNT
	GENERATES	REPORT
	EXTRACTS	REPORT
DEPARTMENT	EMPLOYS	POSITION
	PERFORMS	ACTIVITY
	PREPARES	REPORT
	RECEIVES	REPORT
ACCOUNT	DEBITED-BY	ACTIVITY
	CREDITED-BY	ACTIVITY
REPORT	GENERATED-BY	ACTIVITY
	EXTRACTED-BY	ACTIVITY
	PREPARED-BY	DEPARTMENT
	RECEIVED-BY	DEPARTMENT

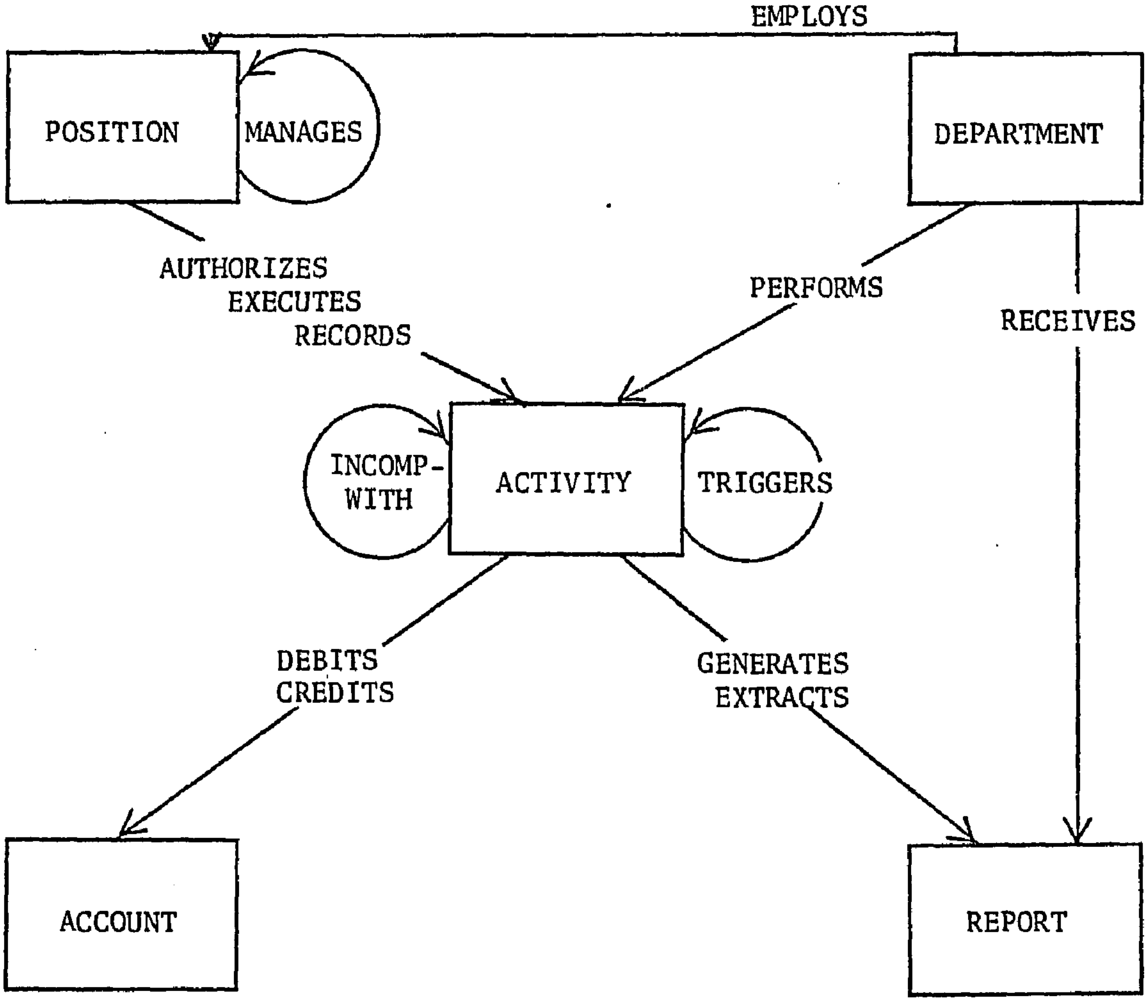


Figure 6. Object-Types and Relation-Types of a Model.

4.1.2 Form of the Model

The model is defined as a series of entries, each consisting of two fields. The first field contains an object-type, attribute-type, or relation-type. The second field contains an object-value (if the first field contains an object-type or a relation-type), or an attribute-value (if the first field contains an attribute-type). The entires are written in groups, in which the first entry of a group consists of an object-type and its value, and all other entries of the group (that is, until another object-type and object-value are defined) are attributes or relations of that first object-value.

All attributes and relations of an object are not necessarily determined at one time. The model may be built up over time. An object may be partially defined at one point in the modeling process and further defined at a later point. Information may be defined redundantly, but the modeler is prevented from defining objects inconsistently (an inconsistent entry, for example, is an object-value having more than one object-type).

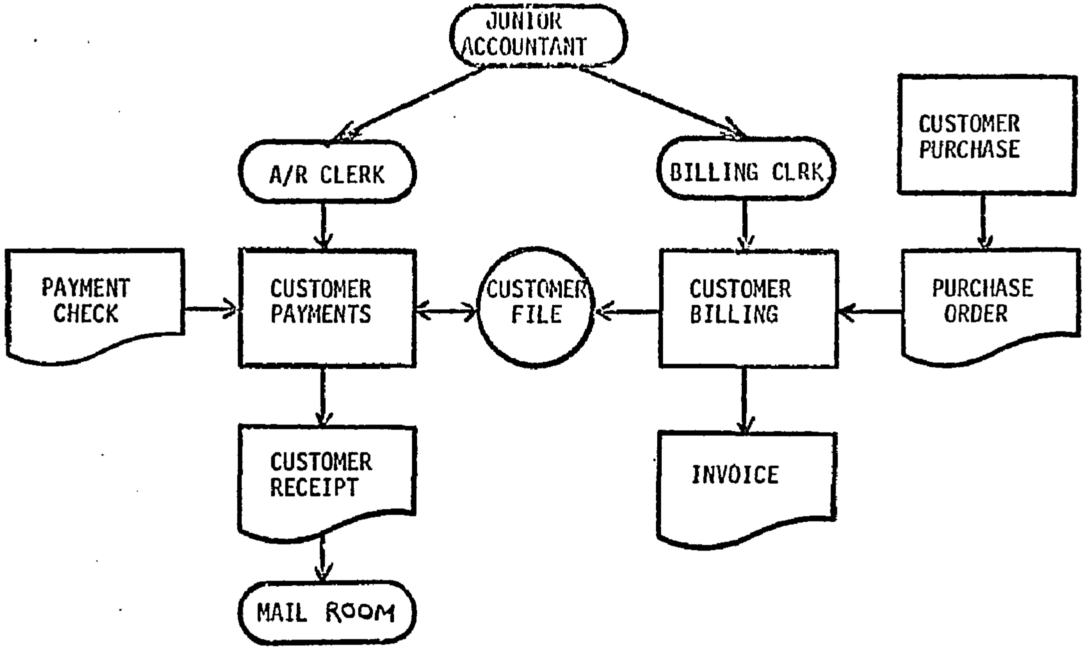
The function of the model is to act as a source of information to be used in an evaluator's decision-making process. Therefore, those aspects of the client that are to be modeled are dictated by the evaluator. In the case of an internal control model, only those aspects of the client which are involved in an evaluation rule (Chapter 5) need to be modeled. For example, if a rule concerns the relationship between departments and reports, then the objects (department and report) and their relations must be modeled; otherwise, they need not be modeled because they are not used. In order to conserve time and costs, it may be stated that information beyond that required should not be modeled. However, this is not necessarily true, for at least two reasons. First, if the model is built based upon the demand of predefined rules, then the later addition of rules may require additional modeling. (That is, the original model is incomplete because the set of rules is incomplete.) Second, indirect relations among objects (e.g., two objects related through a third object) may be omitted if the modeler includes only those objects explicitly appearing in a rule. Therefore, it cannot be specified to what extent the modeler may restrict his or her description of the system of internal control.

Table 3 shows that all relations (with the exception of INCOMPATIBLE-WITH) may be defined in either of two ways. For example, a POSITION AUTHORIZES an ACTIVITY, and an ACTIVITY is AUTHORIZED-BY a POSITION. The modeler may define the relation in either direction; both directions of the relations are implied by either definition and are automatically loaded into the data base.

Figure 7 shows an example of the use of PSL/a. The top portion of the flowchart is an organizational chart. The bottom portion of the flowchart represents the flow of processes and information. Below the flowchart is illustrated how the PSL/a might be written.

4.2 The Modeling Process

The model may be built by the straightforward process of writing PSL/a entries as soon as information is discovered. This process, though, tends to lead to redundancy of information and an incomplete



POSITION	A/R CLERK
PERSON	PETER HUGHS
MANAGED-BY	JUNIOR ACCOUNTANT
EXECUTES	CUSTOMER PAYMENTS
ACTIVITY	CUSTOMER PAYMENTS
INCOMPATIBLE-WITH	CUSTOMER BILLING
EXTRACTS	CUSTOMER FILE
GENERATES	CUSTOMER FILE
EXTRACTS	PAYMENT CHECK
GENERATES	CUSTOMER RECEIPT

REPORT	CUSTOMER RECEIPT
CONTENTS	RECEIPT OF CUSTOMER PAYMENT
RECEIVED-BY	MAIL ROOM
ACTIVITY	CUSTOMER BILLING
EXECUTED-BY	BILLING CLERK
TRIGGERED-BY	CUSTOMER PURCHASE
EXTRACTS	PURCHASE ORDER
GENERATES	INVOICE
GENERATES	CUSTOMER FILE
POSITION	JUNIOR ACCOUNTANT
MANAGES	BILLING CLERK

Figure 7. Illustrative Example of PSL/a.

model (as was learned through the first attempts at modeling with PSL/a). A three-step process is suggested that should yield a more complete, consistent, and accurate model in less time. First, the plan of internal control is flowcharted (a graphical model), and the flowchart is verified for completeness and accuracy. Second, the flowcharted model is transcribed into the PSL/a language, and the model is stored into the data base. Third, completeness and consistency tests are made on the data base using the RULES language (see Section 5.3.2), and the model is corrected.

4.2.1 Flowcharting

A flowchart describing the plan of internal control is prepared. In this form, the modeler can more easily comprehend the entire plan. Completeness of the presentation is more easily determined by viewing such a picture than by trying to comprehend narrative or questionnaire descriptions.

The suggested form of flowcharting is a modified version of that suggested by the American Institute of Accountants (Committee on Auditing Procedure, 1949, pp. 23ff.), and auditing firms (for example, Peat, Marwick, and Mitchell & Co., 1976, Section 2400). The chosen format allows all the constructs of the model to be easily presented directly on the flowchart form.

The flowchart form consists of two parts, a descriptive area and a charting area (refer to Figure 8). In the charting area appears the flowchart. This area is divided into columns, one for each department or functional area of the company. Symbols for objects and reports are

		department	department	department	department
description of activities	position of the person who performs the activity				
DESCRIPTIVE AREA		CHARTING AREA			

Figure 8. Flowchart Format.

drawn in the appropriate column indicating in which department they are performed or generated, and the symbols are connected with flow lines (refer to Figure 9). The name of the activity or report is written within the symbol. If more than one copy of an output is created, one symbol is drawn for each copy, and each symbol is uniquely identified (since each is another output). Attributes of an object are written next to its symbol. Therefore, flow across the form represents flow between departments, while flow down the form represents flow over time.

For each process drawn in the charting area, two entries are written in the descriptive area. First, a more detailed description is made of the activity. Second, the position of the person who performs the activity is entered.

This format is used because it allows all of the PSL/a constructs to be easily presented directly on the flowchart form. Positions are defined in the descriptive area. Activities, accounts, and reports are drawn as symbols in the charting area. Departments are represented as columns of the charting area. Attributes appear next to the appropriate symbols or in the descriptive area. Relations are implied by the flow lines, the position of symbols on the form, and information appearing in the descriptive area.

The use of the flowcharting technique may be observed in the test case appearing in the Appendix.

4.2.2 Transcribing the Model into PSL/a

After flowcharting the plan of internal control, writing the model in the formal PSL/a language is a straightforward process. It is

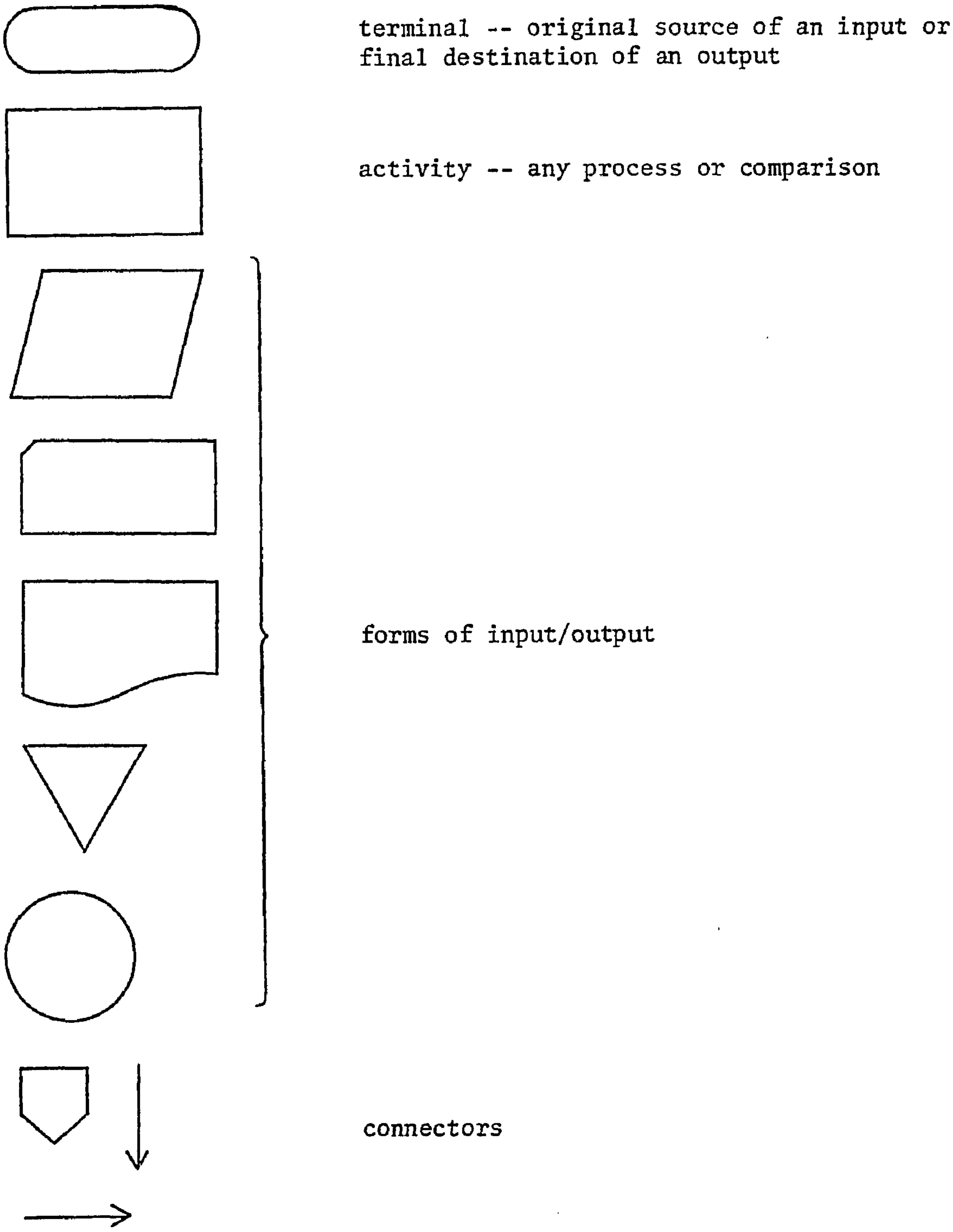

terminal -- original source of an input or final destination of an output
activity -- any process or comparison
forms of input/output
connectors

Figure 9. Flowcharting Symbols.

suggested that the transcription process be ordered in one of three ways: 1) work with all the occurrences of one object-type, and repeat for each object-type; 2) work within one functional area or department at a time, describing all object-types in that area; and 3) describe all aspects of one accounting process at a time; as each object-occurrence is identified, its attributes and relations are defined. By working within a functional area or accounting process, redundant specification of relations may be avoided. All information needed to describe the formal model in PSL/a is on the flowchart form.

4.2.3 Determining Completeness and Consistency of the Model

A model is "complete" if all relevant objects are defined, all attributes of those objects are defined, and all relevant relations of those objects are defined. The completeness of a model cannot be absolutely determined. Building the components of a model is a discovery process, and the description can go only as far as the discoveries; there is no basis or guidelines for determining when all relevant information have been identified. As an example, the only method of determining if all objects have been defined is to compare the defined objects to a list of all possible objects; however, completeness of such a list cannot be determined.

It is desired to have as complete a model as possible, so any technique that aids the discovery is useful. To find missing aspects of a model written in PSL/a, the modeler verifies that all relation-types for each object-occurrence are defined. A missing relation may lead to the discovery of a missing object, and the progression of discoveries grows. A problem exists in that one object may have many occurrences of one relation; initially, the modeler may determine only that at least one relation of each type exists. Most multiple relations are discovered, though, when testing the object at the "other end" of the relation.

A model is "consistent" if each object-value is defined as being of exactly one object-type, and all attribute-values are elements of prescribed sets. An inconsistency would exist, for example, if accounts receivable were defined as both an account and an activity. To prevent inconsistencies, care must be taken at the time of recording the information (building the model). The software that loads the model into the data base will verify consistent usage of object-values.

The internal control model is stored in a data base (as discussed in the next section), allowing computer access to the model. Therefore, testing for completeness and consistency can be programmed. The consistency tests are handled by the data base loader and by user-defined "rules" (refer to Section 5.3.2). Completeness tests also are in the form of user-defined rules that specify conditions that the model must meet. Any incompletenesses or inconsistencies that are discovered by the loader or rules processor are reported to the modeler. The modeler in turn updates the data base containing the model and again processes the rules. This is repeated until the modeler is confident of the completeness and consistency of the contents of the data base.

The automated testing procedure is a major portion of the methodology, and its presentation is delayed until Chapters 5 and 6. A case study illustrating the use of PSL/a is presented in the Appendix.

4.3 The Model Data Base and Data Base Maintenance System

The formal description of the client's plan of internal control, written in PSL/a, is in a computer-interpretable form, but its list-like form does not lend itself to manual evaluation methods. To allow study of the model by computer programs, the model should be stored in such a manner that access to any portion of the model can be readily made, and such that all portions of the model related to a particular object can be readily located.

The model is stored in a data base that provides for "linkages" between related portions of the model. A system of programs loads and edits the data base contents. The data base structure and the maintenance systems are discussed in this section.

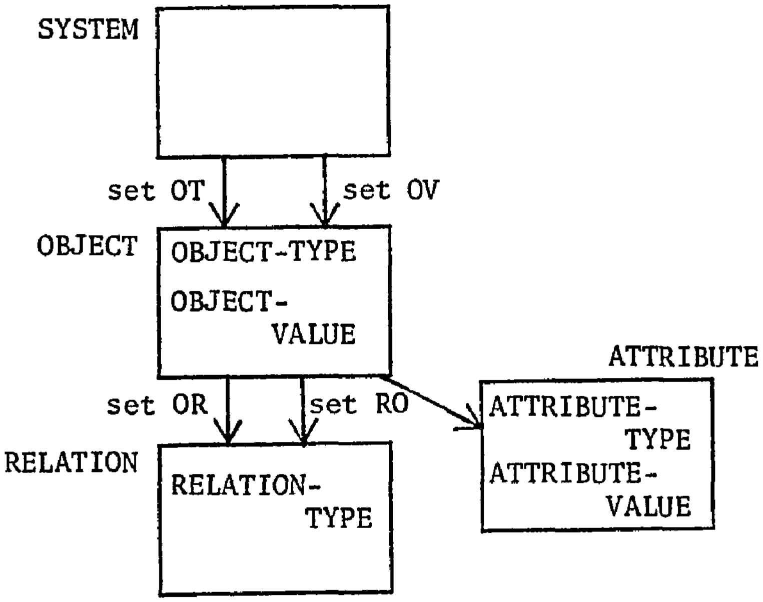
4.3.1 The Model Data Base

The model written in its formal PSL/a format allows only sequential access by the modeler and the computer, because the format is essentially a list. The interrelationships of the components are not easily found in such a format. For computer processing, the representation of the model is changed to make it more accessible by nonsequential methods.

The model is stored into a network data base. A network data base was chosen because of the availability of data base management systems (the chosen data base management system is called ADBMS; Hershey and Messink, 1975); the methodology is not dependent upon the form of the chosen structure. (The algorithm for executing rules, presented in Section 6.4, is dependent upon a network structure. The implications of other structures, for example, relational, are not considered here.) The data base schema is illustrated in Figure 10. It consists of three record types, one for objects, one for attributes of objects, and one for relations between objects.

The OBJECT record-type contains two data items. OBJECT-TYPE is the type of object being described (POSITION, DEPARTMENT, ACTIVITY, ACCOUNT, or REPORT). OBJECT-VALUE specifies a particular occurrence of the OBJECT-TYPE (e.g., president, sales, check credit rating, cash on hand, receipt). There is one occurrence of the OBJECT record for each OBJECT-VALUE of the model. All objects are ordered in two sets. Set OT orders the records on OBJECT-TYPE and set OV orders the records on OBJECT-VALUE, thereby allowing for all values of a particular type or a particular value occurrence to be more easily located.

An ATTRIBUTE record contains the ATTRIBUTE-TYPE (PERSON, STIMULATED, ACCOUNT-TYPE, etc.) and the ATTRIBUTE-VALUE (PETER HUGHS, INTERNALLY, ASSET, etc.). There is one occurrence of the ATTRIBUTE record for each ATTRIBUTE-TYPE identified for a particular occurrence of an OBJECT record. There cannot exist more than one occurrence of an ATTRIBUTE-TYPE for a particular OBJECT occurrence. If the same ATTRIBUTE-TYPE and ATTRIBUTE-VALUE are attributed to several OBJECT occurrences, then there would exist several copies of the ATTRIBUTE record, each linked to its own OBJECT occurrence. The attributes of a



```
RECORD  OBJECT
ITEM    OBTYPE  CHAR    20
ITEM    OBNAME  CHAR    20

RECORD  ATTRIB
ITEM    ATTYPE  CHAR    20
ITEM    ATVALU  CHAR    20

RECORD  RELATE
ITEM    RETYPE  CHAR    20

SET     OT      SORTED  OBTYPE
OWNER   SYSTEM
MEMBER  OBJECT
```

```
SET     OV      SORTED  OBNAME
OWNER   SYSTEM
MEMBER  OBJECT

SET     OR      SORTED  RETYPE
OWNER   OBJECT
MEMBER  RELATE

SET     RO      IMMAT
OWNER   OBJECT
MEMBER  RELATE

SET     OA      SORTED  ATTYPE
OWNER   OBJECT
MEMBER  ATTRIB
```

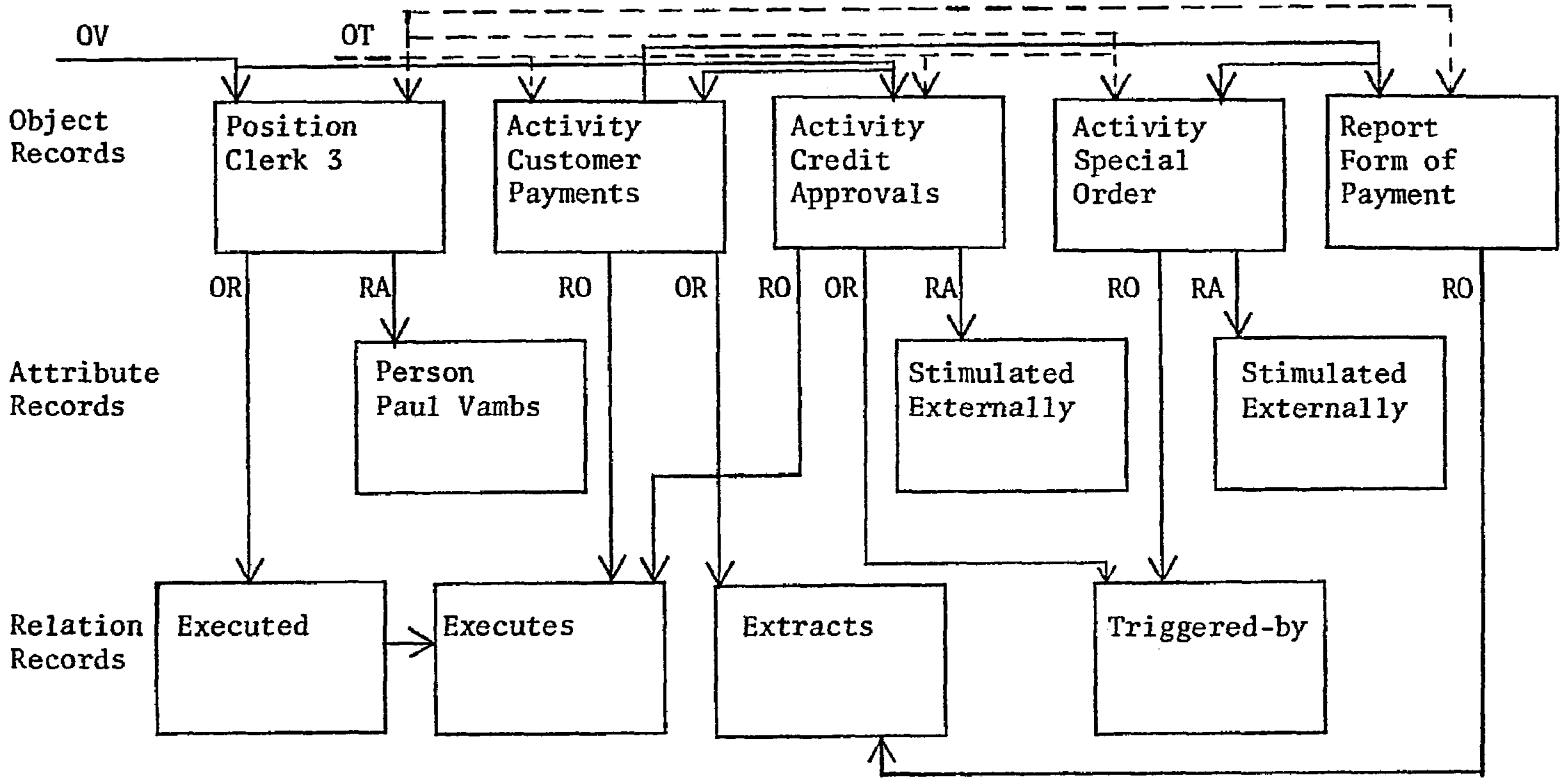
Figure 10. PSL/a Data Base Schema.

particular object are ordered on ATTRIBUTE-TYPE and linked to its OBJECT record through set OA.

A RELATION record contains only the type of relation. It acts as a "nub" record, interfacing two objects. If "person A owns a car," then the OBJECT record for "person A" would be linked to a RELATION record for "own" through set OR; the OBJECT record for "car" would be linked to that RELATION record through set RO. All relations are unidirectional (the car does not own person A) and the direction is indicated by the linkage through the OR and RO sets. (Recall, though, that the opposite relationship will also be stored automatically. When the unidirectional relation "person A owns a car" is stored, the unidirectional relation "a car is owned by person A" is also stored.) Every occurrence of a relation between two objects requires an occurrence of the RELATION record. Set OR is ordered by RELATION-TYPE; set RO is unordered, since there is only one member per owner occurrence.

The requirement for a record-type for RELATION (instead of directly linking objects) and for redundant occurrences of RELATION records is due to restrictions of the data base management system being used. The model may describe logically many-to-many relationships, but they are stored physically as a collection of one-to-one relationships. (The restriction of the data base management system is that a record occurrence may have only one owner in a particular set.)

Figure 11 illustrates the physical occurrences and linkages of a portion of model as it would appear in the data base.



POSITION	CLERK 3	ACTIVITY	CREDIT APPROVAL	ACTIVITY	CUSTOMER PAYMENTS
PERSON	PAUL VAMBS	TRIGGERED-BY	SPECIAL ORDER	EXTRACTS	FORM OF PAYMENT
EXECUTES	CUSTOMER PAYMENTS	STIMULATED	EXTERNALLY	STIMULATED	EXTERNALLY
EXECUTES	CREDIT APPROVALS				

Figure 11. Example of the Physical Linkages within the Data Base.

4.3.2 Data Base Loader

A computer program reads the description of the system of internal control written in PSL/a and builds an equivalent data base. In so doing, some data validation tests are performed.

To recognize legal object-types, attribute-types, and relation-types, a table (LOADER.TAB) is referenced (see Table 4). Because the loader is table-driven, new constructs may be added without altering the software.

When an object-type is declared, the data base is searched to determine if the associated object-value is inconsistent with a previous definition (that is, if the object-value has been previously defined for a different object-type). If no occurrence is found, one is created. This object remains the "primary" object until another object-type is declared. During this time, attributes and relations of the primary object are expected to be defined.

When an attribute is inputed, it is first determined that the attribute-type is valid for the primary object-type by searching LOADER.TAB. It is then determined that the attribute has not already been defined for the primary object occurrence (implying an ambiguity). A new record occurrence is created and linked to the primary object. If it has been previously defined, it is flagged as an error, because the modeler is either trying to load multiple values of a single attribute for an object (which is illegal) or trying to replace the value (perhaps mistakenly). To replace an attribute-value requires an explicit delete (through the use of a maintenance program) and load.

Table 4. Legal Object, Attribute, and Relation Types (LOADER.TAB).

Object-Type	Relation-Type	Related to Object-Type	Inverse Relation-Type	Attribute-Type
POSITION				PERSON
POSITION	MANAGES	POSITION	MANAGED-BY	
POSITION	AUTHORIZES	ACTIVITY	AUTHORIZED-BY	
POSITION	EXECUTES	ACTIVITY	EXECUTED-BY	
POSITION	RECORDS	ACTIVITY	RECORDED-BY	
ACTIVITY				STIMULATED
ACTIVITY				PROCESS
ACTIVITY				PERIOD
ACTIVITY	TRIGGERS	ACTIVITY	TRIGGERED-BY	
ACTIVITY	INCOMPATIBLE-WITH	ACTIVITY	INCOMPATIBLE-WITH	
ACTIVITY	DEBITS	ACCOUNT	DEBITED-BY	
ACTIVITY	CREDITS	ACCOUNT	CREDITED-BY	
ACTIVITY	GENERATES	REPORT	GENERATED-BY	
ACTIVITY	EXTRACTS	REPORT	EXTRACTED-BY	
ACCOUNT				ACCOUNT-TYPE
REPORT				CONTENTS
REPORT				INITIATED
DEPARTMENT	RECEIVES	REPORT	RECEIVED-BY	
DEPARTMENT	PREPARES	REPORT	PREPARED-BY	
DEPARTMENT	PERFORMS	ACTIVITY	PERFORMED-BY	
DEPARTMENT	EMPLOYS	POSITION	EMPLOYED-BY	

Loading a relation requires many tests. When defining a relations, the modeler does not explicitly state the object-type of the object-value to which the primary object is related (for example, in POSITION A/R CLERK / EXECUTES CUSTOMER PAYMENTS, the object-type of CUSTOMER PAYMENTS is only implied to be ACTIVITY). Therefore, the object value is first located in the data base and its object-type fetched. Only then can it be determined (through a table look-up) that the relation is valid between the two object-types. It must also be determined that the relation between the two object-values has not been previously loaded, because storing redundant relations would later lead to inconsistencies in searching the data base. If the second object-value cannot be located, then a new object-record is created, the object-type being implied (according to the LOADER.TAB) by the relation. Finally, since all relations are stored unidirectionally, the inverse relation (e.g., MANAGES/MANAGED-BY) is loaded by the loader program in the same manner. (The inverse relation is also located in LOADER.TAB).

4.3.3 Data Base Utilities

Once created, the data base must be updated, corrected, reorganized, and viewed.

Adding new data requires the use of the loader program. To replace data, the original occurrences must be deleted and the new occurrences loaded.

A maintenance package handles deleting and housekeeping. To delete an attribute, the user specifies the object-type, object-value, and attribute-type. When the attribute record is deleted, the user is informed of the removed attribute-value. To delete a relation, the user must specify the primary object-type, its object-value, relation, secondary object-type, its object-value, and the inverse relation. The user is notified of any errors or absent data.

It is possible for the data base to contain objects which have no attributes or relations. This may occur during loading or during the deletion process. The user may request the utility program to search for such "loner" objects (object records with no attributes and no relations). For each one found, the user is asked if it should be deleted from the data base.

Another utility is available for dumping the contents of the data base. There are two forms of the listing. The first is a detailed listing of the data, including the set names, record names, and physical data base keys. The second is an ordered listing (ordered by either object-type or object-value) of the data. For each object-value, its attributes and relations are detailed. This output is in the same format as that of input data to be loaded, providing a comprehensible description of all objects (and serving as a good form of data base backup).

4.4 Spinoffs of the Modeling Process

Separate from the methodology of an automated evaluation, the modeling language, procedure, and subsystem offer contributions beyond those originally sought.

4.4.1 Documentation

The PSL/a language and data base system offer an organized procedure for collecting, arranging, and maintaining documentation of systems of internal control. The documentation is not limited to use by external auditors or to the application of internal control. PSL/a may be used to describe any situation involving the interactions of objects and activities. An example of another application area is the process flow of a manufactured item, showing the interrelations of subassemblies.

4.4.2 Comprehension

Once the model is built, access to any portion of the model, or documentation, is easily managed. All relevant information may be quickly located. The data base is a medium for storing the model, or documentation, so several evaluators may view the data in different respects. Such a structured documentation removes bias and ambiguities from the writing style of the documentation, and evaluations may be made more consistently.

4.4.3 Evaluation

While building the model, the modeler is forced to search for certain characteristics and relations of the client. In so doing, evaluation processes are being simultaneously conducted, even though the modeler might not be cognizant of this. For example, the modeler is guided to record a particular relation of an object; if that relation cannot be located, then an evaluation process is triggered. Through an organized method of data collection, then, the evaluation process begins before the purposeful evaluation.

4.4.4 Planning

After a model has been built, it may be altered to study the effect of change. In this manner, the formalized documentation system may be used as a planning tool. For example, if a weakness is found in the system of internal control, the effect of adding a person or a process could be studied to see if the weakness can be remedied. PSL/a may also be used as a planning tool for the initial design of a system (as opposed to modifying an already existent design). In this case, the model would be built as it is desired to be implemented. Tests are made on the system to discover weaknesses before these weaknesses are ever implemented.

CHAPTER 5

FORMALIZATION OF EVALUATION CRITERIA

The methodology for automating an evaluation of the plan of internal control relies on two sets of inputs. The first (the topic of Chapter 4) is a description of that which is to be evaluated. The second (the topic of this chapter) is a representation of the criteria upon which the auditor bases the evaluation. The evaluation criteria are rules that describe the attributes of a good plan of internal control and that are to be applied to the data base description of an extant plan of internal control.

5.1 Use of Rules for Evaluation Processes

To automate an evaluation, the methods of the auditor must be expressed to the computer. There are several techniques for expressing these. One manner requires the auditor to build a model (using PSL/a) of an ideal plan of internal control, and then to compare the ideal with the actual, noting all differences. However, there is no single ideal arrangement of object, attributes, and relations; a comparison of even two ideal plans would yield a great number of differences.

The manner of expressing the auditor's evaluation criteria chosen for this methodology is through the definition of a set of "rules." Each rule describes required or disallowed objects, attributes, and relations of a plan of internal control. Because this is the chosen function of the rules, the syntax of the rules language was developed over time to allow such expressions. Each rule is processed against the entire model data base (the process is discussed in Chapter 6), being applied to every object, attribute, or relation which is a concern of the rule.

5.2 Description of the RULES Language

The rules are expressed in a formal language called RULES. The set of rules comprises a sequential file and may be executed in batch or interactive mode. The operators of RULES are summarized in Table 5. The interpreter which processes the rules is table-driven, so the language processor allows for the use of noise words and synonyms of operators and operands. For example, "FOR EACH" is interpreted as the operator "@" ("FOR" is a noise word and is ignored; "EACH" is a synonym of "@"). The explicit syntax of RULES is presented in Table 6 and Figure 12.

There are three general forms of rules. The first type will be described extensively; the other two are variations of the first and will be briefly described.

5.2.1 RULE-TYPE 1: FOR EACH OCCURRENCE OF <CLAUSE> THERE MUST EXIST <CLAUSE>

In general, a rule expresses a search pattern for the model data base. All data of the data base are of the form: object related-to object related-to etc. Such a list (e.g., position authorizes a transaction which debits an account) is called a "string." A "clause" is a string or a collection of strings and appropriate operators. A rule consists of strings and operators.

Table 5. Operators of the RULES Language.

@	"(for) each occurrence of"
∃	"there must exist"
∄	"there must not exist"
1	"1 and only 1 occurrence of"
∈	"must be an element of the set"
∉	"must not be an element of the set"
=	"all the (concatenated) values to the left must equal all the (concatenated) values to the right"
≠	"all the (concatenated) values to the left must not equal all the (concatenated) values to the right"
S=	"equality of values among an occurrence of each of the following elements"
S≠	"inequality of values among an occurrence of all of the following elements"
Λ	"AND"
V	"OR"
SΛ	"an occurrence of each of the following elements"
SV	"an occurrence of any of the following elements"
!	"repeated 0 or more times"
" "	literal
[ ]	"the value of"
( )	used for logical groupings
.	name extension
,	list separator
$	end of rule

Table 6. BNF Syntax of the RULES Language.

<primitive>::=	<object>|<attribute>|<relation>| <null>|<character-string>
<object-prim>::=	<object>|<object><attribute> [<object-prim>]
<relation-prim>::=	<relation>
<string-1n>::=	<object-prim>|<null>
<string-2>::=	<object-prim><relation-prim>| <object-prim><relation-prim><string-2>
<string-2n>::=	<string-2>|<null>
<string-3>::=	<string-2><string-1n>| <string-2n><object-prim>
<string-3n>::=	<string-3>|<null>
<string-4>::=	<relation-prim><object-prim>| <relation-prim><object-prim><string-4>
<string-4n>::=	<string-4>|<null>
<ellipsis-string>::=	(<string-2n>(<string-2>)!<string-3n>)| (<string-2n><object-prim>(<string-4>)!<string-4n>
<string>::=	<string-3>|<ellipsis-string>
<literal>::=	"<character-string>"
<value-string>::=	<string>|<literal>
<simple-op>::=	=|≠|∧|∨
<set-op>::=	S=|S≠|S∧|S∨
<set>::=	<value-string>,<value-string>| <value-string>,<set>
<clause>::=	<string>| <value-string><simple-op><value-string>| (<clause>)<simple-op>(<clause>)| <set-op>(<set>)

Table 6, Continued.

<occurrence-op1>::=	@|1|SOME
<occurrence-op2>::=	1|<null>
<existence-op>::=	∃|∄
<element-op>::=	∈|∉
<rule-1>::=	<occurrence-op1><clause><existence-op> <occurrence-op2><clause>$
<rule-2>::=	<existence-op><occurrence-op2><clause>$
<rule-3>::=	<occurrence-op1><clause><element-op>(<set>)$
<rule>::=	<rule-1>|<rule-2>|<rule-3>

The three rules may be summarized as:

<rule-1>	for each occurrence of ... there must exist ...
<rule-2>	there must exist ...
<rule-3>	each occurrence of ... must be an element of the set ...

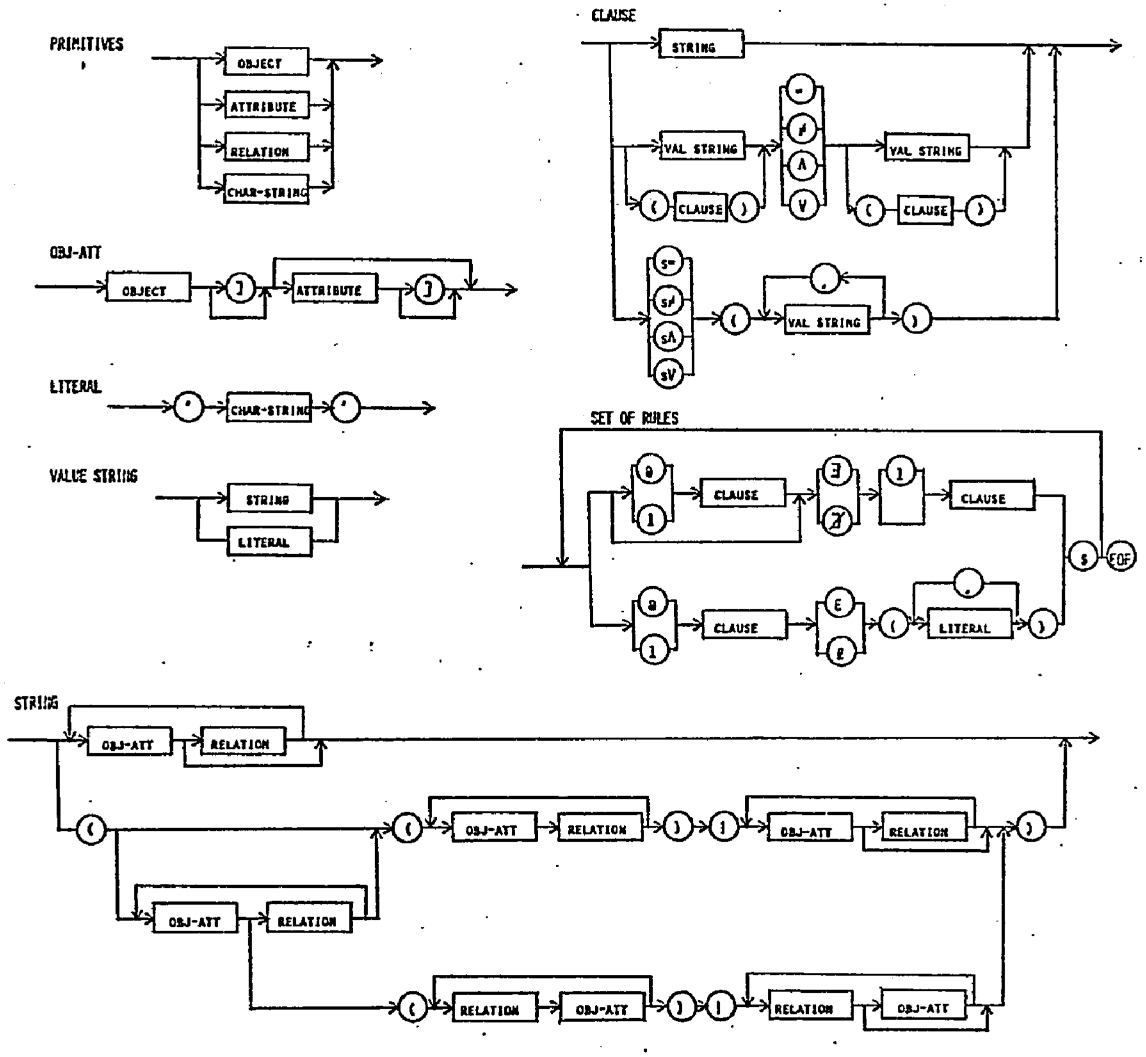


Figure 12. Syntax of the RULES Language.

During execution of a rule, an occurrence of the first clause is searched for in the data base. If it is found, it then acts as a qualifier for the second clause which is searched for next. If an occurrence of the second clause cannot be found, a "violation" of the rule is indicated. This process is continued for each occurrence of the first clause which can be found within the data base.

For example, examine FOR EACH OCCURRENCE OF POSITION THERE MUST EXIST POSITION AUTHORIZES TRANSACTION $. The string, POSITION, is the first clause. The string, POSITION AUTHORIZES TRANSACTION, is the second clause. In "executing" the rule, a search for an occurrence of POSITION will be performed. If an occurrence is found, POSITION AUTHORIZES TRANSACTION will be searched, where POSITION of the first clause is the <u>same occurrence</u> of POSITION in the second clause. After the first pass, another occurrence of POSITION (for the first clause) will be searched. This will continue until "EACH OCCURRENCE OF POSITION" has been tested. (In this rule, FOR, OCCURRENCE, OF, THERE, and MUST are noise words; EACH and EXIST are synonyms for the operators @ and ∃, respectively.)

If a clause consists of only a string, then the rule is searching for <u>existence</u> of the string. However, a clause also may test equality of values of strings and may include logical ANDs and ORs.

The instructions given below describe the legal use of the RULES language for rule-type 1. Instructions 1 through 5 explain the formation of a complete rule:

1. Rule syntax. A rule consists of two parts -- qualification and existence. The form of a complete rule (of rule-type 1) is: <qualification operator> <clause> <existence operator> <clause> <end-of-rule operator>.

2. @, 1, SOME, LIST. The qualification operators are: @, 1, SOME, and LIST. @ means "for each occurrence of the following clause." During evaluation of the rule, an occurrence of the clause will be found, and the existence part will be executed. Then another occurrence of the first clause will be found before executing the existence part of the rule. This will be repeated for each occurrence of the first clause. If the result of execution of the existence part is ever "false," then the data base values associated with the first clause are printed as a violation report.

   1 means "for one and only one occurrence of the following clause." If no occurrences or more than one occurrence of the first clause is found for which the result of execution of the second part is "true," then a violation is reported.

   SOME means "at least one occurrence of the following clause." Execution of the rule terminates after any occurrence of the first clause is found for which the second clause is satisfied.

   LIST is executed in the same manner as @. However, instead of reporting violations, all cases which satisfy the rule are reported. With the LIST operator, the existence part of the rule is not required. For example, LIST POSITION RECORDS $ will list all positions which record something.

3. ∃, ∄, ∃1, ∄1. The existence operators are ∃, ∄, ∃1, and ∄1. ∃ means "there must exist at least one occurrence of the following clause." ∄ means "there must not exist any occurrences of the following clause." ∃1 (∄1) means "there must (not) exist exactly one occurrence of the following clause."

4. $. The end-of-rule operator is $.

5. Free-format. All rules are specified in free-format. Each line is limited to eighty character positions. Anything past the eightieth column is ignored without warning. Spaces may appear between operators and operands, but are required only to separate words of a string. (Tabs are not allowed.)

Instructions 6 through 10 involve the formation of a string:

6. String. A string is of the form OBJECT RELATION OBJECT RELATION etc. The string must start with an OBJECT and may end with any type of operand. Attributes may exist in a string and are placed immediately following their objects; e.g., in POSITION PERSON AUTHORIZES ACTIVITY, POSITION and ACTIVITY are objects, PERSON is an attribute of POSITION, and AUTHORIZES is a relation. When searching for an occurrence of this string, one will be found only for a POSITION that has the PERSON attribute declared, and that AUTHORIZES an ACTIVITY.)

7. Name extension. Every instance of an object-type in a rule refers to the same occurrence in the data base. For example,

POSITION MANAGES POSITION refers to any occurrence of POSITION managing that same occurrence of POSITION (a position managing itself). A name extension consisting of a decimal point and a digit may be appended to an object-type or attribute-type to specify different occurrences while searching. In this case, like object-types with equal name extensions will result in the search for the same occurrence in the data base; like object-types with unequal name extensions will result in the search for different occurrences in the data base. For example, POSITION.1 MANAGES POSITION.2 will search for a position managing a different position. POSITION.1 MANAGES POSITION.2 EXECUTES ACTIVITY RECORDED-BY POSITION.1 will search for a position that manages a different position that, in turn, executes some activity that is recorded by the first position.

8. [ ]. The value of an object or an attribute may be needed for comparisons to literals or to other values. If an object or attribute is enclosed in brackets, [ ], its value will be extracted from the data base and retained for later comparisons. The brackets are used in conjunction with the operators, =, ≠, $\{$, and $\not\{$, and provide a means of further qualifying a construct occurrence.

A value is twenty characters long, left-justified, blank-filled. If more than one value is indicated in a string, they are concatenated, twenty characters each, to a maximum of five values (100 characters). If the values of two contiguous words of a string are desired, they must each be enclosed in brackets; the two words cannot be grouped in one set of brackets. (Actually, the left bracket is treated as a "noise word," and the right bracket is the value operator.)

[ ] may not be used if the object has a name extension (e.g., [POSITION.1] is illegal).

9. " ". A string may alternatively consist of one literal. The literal is enclosed in (double) quotes. If the literal consists of less than twenty characters, it will be left-justified and blank-filled. If it is greater than twenty characters, it will be truncated after the leftmost twenty characters. The quotes are used in conjunction with the operators, =, ≠, {, and {̸.

10. !. The ellipsis operator, !, may be applied to a substring (a portion of a string). The substring is placed in parentheses and followed by the ! operator; then the entire string must be placed in another set of parentheses. The substring must either start with an object and end with a relation or start with a relation and end with an object. For example, @(POSITION.1 (MANAGES POSITION.2)! AUTHORIZES) } ... $. Its effect is to cause the substring to occur zero or more times. The entire string is handled as if the operand of the ellipsis operator does not exist, then as if it exists once, then as if it exists twice by looping the operand, and iterates as many times as possible (i.e., @ POSITION.1 MANAGES POSITION.2 MANAGES POSITION.3 ... MANAGES POSITION.N AUTHORIZES }). When interpreting a rule

containing the ellipsis, the substring will iterate as many times as necessary in order to satisfy the rule, or until an occurrence cannot be found for the repeated substring. The example rule may be considered to be an abbreviated form of the rule, @(POSITION.1 AUTHORIZES) V (POSITION.1 MANAGES POSITION.2 AUTHORIZES) V (POSITION.1 MANAGES POSITION.2 MANAGES POSITION.3 AUTHORIZES) V ... } ... $. (If an ellipsis operand has a name extension, such as POSITION.2, the name extension is global over the entire rule, but does not refer to previous iterations of the operand.)

Care must be taken in the use of the ellipsis operator. Although the substring may begin with an object or a relation, the looping process must be logically correct; the last word of the substring must lead into the first word of the substring. It should be kept in mind that, for its first interpretation, the substring is ignored (repeated zero times). Also, iteration ceases upon a nonexistence condition; it is possible for nonexistence of the substring on four iterations, and yet it may exist with five iterations (the five-iteration case would not be discovered).

Instructions 11 through 15 involve operations among strings to form a clause:

11. =, ≠. The equality or inequality of the values of two entities may be declared as a qualifier to a rule. For example, @ [POSITION] = "PRESIDENT" } POSITION MANAGED-BY $. The first clause concerns all positions except that of the president. (The

rule states that everyone, except the president, must have a manager.) Each of the entities being compared must have a value operator, [ ], or must be a literal.

12. S=, S≠. The equality of more than two strings may be determined by creating a set of strings to be evaluated. The form is S=(string,string,string,string). There may be as many as five strings, separated from one another with a comma, completely enclosed in one set of parentheses, and preceded by S= or S≠. Each string must contain the value operator, [ ], and one string may be a literal. S= requires that the values extracted for all strings must be equal. S≠ requires the values extracted for all the strings must be unique. In the case of the nonexistence of a string, the value of that string is considered to be unequal from the values of all other strings.

13. Λ, V. The logical AND, Λ, and the logical OR, V, may be applied to the existence of two strings. For example, POSITION AUTHORIZES Λ POSITION EXECUTES will be evaluated as "true" if the position both authorizes and executes activities. POSITION AUTHORIZES V POSITION EXECUTES will be evaluated as "true" if the position authorizes, or executes, or authorizes and executes activities. Also, one or both operands may involve an equality operator, = or ≠. In this case, the operand involving the equality must be enclosed within parentheses. For example, POSITION AUTHORIZES V (POSITION EXECUTES [ACTIVITY] ≠ "SALES

TRANSACTION"). Also, Λ and V expressions may be nested within one another through the use of parentheses.

If V is used in the first clause, the operands must not involve equality operators.

14. SΛ, SV. The existence of several strings may be determined with the set of operators, SΛ and SV. The general form is SΛ (string, string, string, string). There may be any number of strings, separated from one another with a comma, completely enclosed in one set of parentheses, and preceded by SΛ or SV. Each element of the set may involve the equality of two strings instead of just the existence of one. SΛ requires the existence of all strings. SV requires the existence of at least one string.

    If SV is used in the first clause, the operands must not involve equality operators.

15. Right-to-left. Unless otherwise grouped within parentheses, the =, ≠, Λ, and V operators are evaluated from right to left with the same level of precedence. A Λ [B] = [C] is interpreted as A Λ ([B] = [C]). [A] = [B] V C is interpreted as [A] = ([B] V C); it will determine if B or C exists and then compare the value of A to that of B, and it is probably illogical.

5.2.2 RULE-TYPE 2: THERE MUST EXIST <CLAUSE> $

This is the same as rule-type 1, without its first part ("for each occurrence of <clause>"). Its meaning is self-explanatory.

5.2.3 RULE-TYPE 3: EACH OCCURRENCE OF <CLAUSE> MUST BE AN ELEMENT OF THE SET (<VALUES>) $

The rule will test particular values in the data base to see if they are among a set of given values. The first part of the rule is the same as that for rule-type 1. The first part is interpreted as "each occurrence of the following clause." The clause must contain the value operator, [ ].

The second part of the rule, interpreted as "must (not) be an element of the following set," consists of the operator ∈ or ∉, followed by a set of literal values, each separated by commas and completely enclosed within one set of parentheses. For example, @ ACCOUNT [ACCOUNT-TYPE] ∈ ("ASSET", "LIABILITY", "EQUITY") $ (here, ACCOUNT-TYPE is an attribute of ACCOUNT).

This rule-type is especially useful for determining the consistent application of attribute values.

5.3 Application of RULES

A rule expresses a search pattern to be applied to the model data base. As such it can be used as a tool for a function where searching is required. The primary purpose of the rules has already been established as to represent the auditor's criteria for evaluating the plan of internal control. It was indicated in Section 4.2.3 that the rules can also be used to help determine completeness and consistency of the model as part of the modeling process. Both of these uses are the subjects of the following two subsections.

5.3.1 Rules for Expressing Auditing Criteria

Examples of the types of rules that the auditor could use for evaluating the internal control plan are listed below. No claim is made that this set is complete. The auditing rules must be a creation of the auditor, because the rules represent the auditor's personal evaluation criteria. When first using RULES, the set is expected to be rather limited. Over time, this set will grow as the auditor gains experience with its use and learns to specify previously unwritten rules.

1. Each activity should be authorized by one position, executed by one position, recorded by one position, and performed by one department:

```
@ ACTIVITY } 1 SΛ(ACTIVITY AUTHORIZED-BY,
                  ACTIVITY EXECUTED-BY,
                  ACTIVITY RECORDED-BY,
                  ACTIVITY PERFORMED-BY) $
```

2. For each accounting activity, three different persons should be responsible for authorization, execution, and recording:

```
@ ACTIVITY }
            S≠ (ACTIVITY AUTHORIZED-BY POSITION [PERSON],
                ACTIVITY EXECUTED-BY POSITION [PERSON],
                ACTIVITY RECORDED-BY POSITION [PERSON]) $
```

3. The responsibilities of execution and recording of transactions should not be performed in the same department:

```
    @   ACTIVITY EXECUTED-BY POSITION.1
      Λ ACTIVITY RECORDED-BY POSITION.2 ʒ
              DEPARTMENT EMPLOYS POSITION.1
            Λ DEPARTMENT EMPLOYS POSITION.2 $
```

4. Two activities which are incompatible must not involve the same positions:

```
@ ACTIVITY.1 INCOMPATIBLE-WITH ACTIVITY.2 ʒ
    SV (ACTIVITY.1 AUTHORIZED-BY POSITION AUTHORIZES ACTIVITY.2,
        ACTIVITY.1 AUTHORIZED-BY POSITION EXECUTES ACTIVITY.2,
        ACTIVITY.1 AUTHORIZED-BY POSITION RECORDS ACTIVITY.2,
        ACTIVITY.1 EXECUTED-BY POSITION AUTHORIZES ACTIVITY.2,
        ACTIVITY.1 EXECUTED-BY POSITION EXECUTES ACTIVITY.2,
        ACTIVITY.1 EXECUTED-BY POSITION RECORDS ACTIVITY.2,
        ACTIVITY.1 RECORDED-BY POSITION AUTHORIZES ACTIVITY.2,
        ACTIVITY.1 RECORDED-BY POSITION EXECUTES ACTIVITY.2,
        ACTIVITY.1 RECORDED-BY POSITION RECORDS ACTIVITY.2) $
```

5. A hierarchical organizational chart requires that no position may manage itself:

```
@ POSITION.1 ʒ
        (POSITION.1 (MANAGES POSITION.2)! MANAGES POSITION.1) $
```

6. Concerning a chain of activities which are triggered by one another, a person who authorizes any of the activities may not execute or record <u>any</u> of the activities in the chain:

```
@ ACTIVITY.1 (TRIGGERS ACTIVITY.2)! TRIGGERS ACTIVITY.3 ∃
        S≠ (ACTIVITY.1 AUTHORIZED-BY POSITION [PERSON],
            ACTIVITY.2 EXECUTED-BY POSITION [PERSON],
            ACTIVITY.3 RECORDED-BY POSITION [PERSON]) $
```

7. No position may manage itself:

```
@ POSITION.1 ∄
   (POSITION.1 (MANAGES POSITION.2)! MANAGES POSITION.1) $
```

5.3.2 Rules for Determining Completeness and Consistency of the Model

Completeness of a model (refer to Section 4.2.3 for a discussion of the definition of completeness) may be determined to the extent that an occurrence of each type of attribute and relation applying to each object occurrence has been defined at least once. Consistency of a model (again, refer to Section 4.2.3) may be determined by examining the application of attribute values and certain interrelations of related objects. A collection of rules for aiding the determination of completeness and consistency is now presented.

1. Each position should have one manager:

```
@ POSITION ∃ 1 POSITION MANAGED-BY $
```

2. Each position should either authorize, execute, or record an activity:

```
@ POSITION ∃ SV(POSITION AUTHORIZES,
                POSITION EXECUTES,
                POSITION RECORDS) $
```

3. Each position should be assigned to one department:

```
@ POSITION ∃ 1 POSITION EMPLOYED-BY $
```

4. The attribute of each position should be defined:

```
@ POSITION ∃ POSITION PERSON $
```

5. All relation-types of each activity must be defined:

```
@ ACTIVITY ∃ SΛ(ACTIVITY AUTHORIZED-BY,
               ACTIVITY EXECUTED-BY,
               ACTIVITY RECORDED-BY,
               ACTIVITY TRIGGERS,
               ACTIVITY TRIGGERED-BY,
               ACTIVITY PERFORMED-BY,
               ACTIVITY GENERATES,
               ACTIVITY EXTRACTS) $
```

6. The attribute of activities must be consistently defined:

```
@ ACTIVITY [STIMULATED] ∈ ("INTERNALLY", "EXTERNALLY") $
@ ACTIVITY [STIMULATED] = "EXTERNALLY"
      ∄ ACTIVITY TRIGGERED-BY $
@ ACTIVITY [STIMULATED] = "INTERNALLY"
      ∃ ACTIVITY TRIGGERED-BY $
```

7. Each department which receives a report should use it in an activity:

```
@ DEPARTMENT RECEIVES REPORT ∃
      DEPARTMENT PERFORMS ACTIVITY EXTRACTS REPORT $
```

8. All account-types must be defined and must be defined consistently:

```
@ ACCOUNT ∋ [ACCOUNT-TYPE] ≠ "   " $

@ ACCOUNT [ACCOUNT-TYPE] ∈ ("ASSET", "LIABILITY", "EQUITY") $
```

9. Each account must be debited and credited by activities:

```
@ ACCOUNT ∋ ACCOUNT DEBITED-BY ∧ ACCOUNT CREDITED-BY $
```

10. If an activity debits an account, then it must credit an account, or it must trigger another activity which credits an account (and vice-versa):

```
@ ACTIVITY.1 DEBITS ∋

        (ACTIVITY.1 (TRIGGERS ACTIVITY.2)! CREDITS) V

        (ACTIVITY.1 (TRIGGERED-BY ACTIVITY.2)! CREDITS) $

@ ACTIVITY.1 CREDITS ∋

        (ACTIVITY.1 (TRIGGERS ACTIVITY.2)! DEBITS) V

        (ACTIVITY.1 (TRIGGERED-BY ACTIVITY.2)! DEBITS) $
```

11. All relation-types of each report must be defined:

```
@ REPORT ∋ SΛ(REPORT GENERATED-BY,

              REPORT EXTRACTED-BY,

              REPORT PREPARED-BY,

              REPORT RECEIVED-BY) $
```

5.4 Sufficiency of a Set of Rules and of the RULES Language

Defining the set of rules for evaluating the plan of internal control is the responsibility of the auditor. The auditor must determine aspects of the plan that should be tested, and then express the tests in

RULES. Initially, these rules are derived from generally accepted auditing practices. The collection of rules will grow over time as the auditor continues his or her experience with the system. The discovery of a violation of a rule (that is, a weakness in internal control) prompts the auditor to generate additional rules. This iterative discovery process is the same whether the audit manner be manual or performed by computer.

Sufficiency of a set of rules can be determined only by the auditor's judgment. Sufficiency changes from one audit case to another. The auditor implicitly declares the rules to be sufficient once he or she is satisfied with the resulting evaluation reports.

Sufficiency of the RULES language, as is the case for any language, is determined by the user. If the user is able to express all desired commands with the language, then the language is considered sufficient. Sufficiency can be determined only empirically, and through time the RULES language is expected to expand (Section 8.1.2 proposes some of these extensions).

CHAPTER 6

EXECUTION OF THE AUTOMATED EVALUATION

A description of accounting "objects," "attributes," and "relations" to be evaluated has been made computer-accessible via PSL/a (Chapter 4). A representation of the auditor's evaluation criteria has been made computer-interpretable via RULES (Chapter 5). A computer software system can now process the evaluation criteria (the rules) against the plan of internal control (the model). In so doing, aspects of the model that violate a rule are described as the output of the system. The evaluation software is called ICE (Internal Control Evaluator).

Once the model of the plan of internal control has been loaded into the data base, the rules are read and processed, one at a time (at this stage of the research, all rules are independent of one another). After the rule is determined to be syntactically correct, the data base is searched according to the rule. If an aspect of the data base violates the specification of the rule, a violation report is generated that describes the error condition. This process is repeated for each rule.

The definition of an explicit algorithm for how the rules must be executed is not a goal of this dissertation; the actual implementation method is a function of the implementor. However, an overview of the algorithm used for the research implementation is presented in this chapter.

6.1 Logging on to the ICE System

ICE, in interactive mode, prompts the user for three file names -- the model data base file, the rules input file, and the violation output file. The rules may exist in a sequential file, or the user may enter them online at the terminal. The violation reports may be sent to a sequential file or to the terminal. Therefore, the user may operate the system in an interactive mode (entering a rule, observing the violations, and entering additional rules) or in a batch mode (passing to it all rules and then observing all violations).

6.2 Scanning the Rule

A rule is read and copied to the violation report file and the terminal. As ICE scans the rule, each word (except literals, which are enclosed in quotes) is looked up in a table. This table contains all legal operators, object-types, attribute-types, relation-types, synonyms for all operators and operands, and noise words. Each word has associated with it a code indicating its type. If a word of the rule cannot be located in the table, the user is prompted for one of four responses: 1) it is an error -- the user re-enters the word; 2) it is a noise word -- the word is ignored this time; 3) it is a permanent noise word -- the word is added to the table and identified as a noise word to be ignored in all future uses; and 4) it is a new synonym (or operator) -- the word of which this is a synonym is entered, along with its identifying code.

6.3 Parsing the Rule

The parsing process uses three stacks: a temporary stack holds operators until they are ready to be placed onto the operator stack; the operator stack stores in postfix order the operators and pointers to their operands; and the operand stack holds all operand strings (objects, attributes, relations, and literals).

The form of a rule has three peculiarities that result in complicating the parsing and execution processes. First, an operand is a string of words (objects, attributes, and relations, or a literal) and may be of any length. Second, some operators are unary (@, }, {, $, !,]), some are binary (Λ, V, =, ≠), and some have a variable number of operands (SΛ, SV, S=, S≠). Third, some operators precede their operands (@, }, {, SΛ, SV, S=, S≠), some follow their operands (!,], $), and some separate their operands (Λ, V, =, ≠). Because the number of operands for an operator may vary, and because the length of an operand may vary (since it may be a "string" (refer to Section 5.2), the operands are maintained in a separate stack. With each word of the operand is stored its type and its name extension (e.g., for POSITION.2, the type is OBJECT and its name extension is "2"). As an operand string is stored in the operand stack, a temporary pointer is used to indicate the location of the beginning of the operand. This pointer will be stored in one of the other two stacks with its associated operator (as explained later).

To simplify the "execution" of the rule, the rule is parsed into postfix order. Each operator is assigned a precedence value. When an operator is scanned, its precedence value is compared with that of the latest operator on the temporary stack. If the newly scanned operator is of greater precedence than that in the temporary stack, the operator is immediately pushed onto the final operator stack. If it is of equal precedence, it is pushed onto the temporary stack. If it is of lesser precedence, operators are transferred from the temporary stack to the final operator stack, until an operator is found of lesser or equal precedence. Because the order of operators to their operands is not consistent (some precede, some follow, and some separate their operands), special cases are recognized and stacked appropriately.

As any operator is pushed onto a stack, it is accompanied by a pointer to the beginning of its operand(s) in the operand stack. If the operator has multiple operands, one pointer for each operand is pushed onto the operator stack before the operator itself is pushed. Operands appear to be located in the operator stack as expected when parsing into postfix order; however, these "operands" are actually pointers to the operand values in the operand stack. Because the span of control (the inclusion of other operators and operands) varies among usages of the operators, another pointer is stored with each operator. This pointer links back into the operator stack indicating the beginning of the operator's span of control.

An example of the results of parsing a rule appears as Figure 13. The PARSER algorithm is shown in Figure 14.

6.4 Execution of a Rule

The basic operation performed during "execution" of a rule is the FIND. The FIND attempts to find an occurrence of a string (either a

@ POSITION.1 ⁊ POSITION.1 MANAGES (POSITION.2 MANAGES)! POSITION.1 $

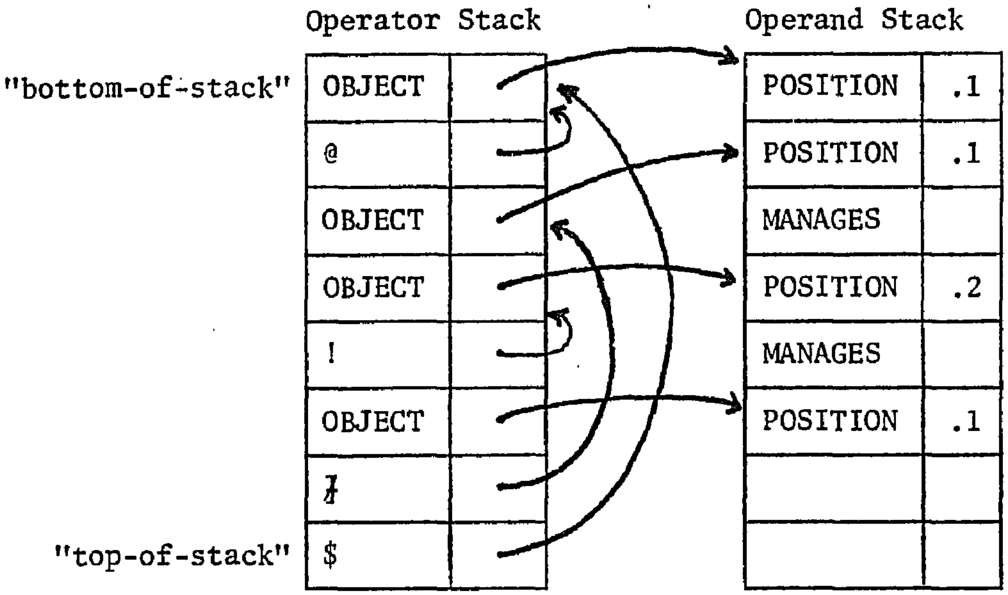


Figure 13. Stacks Resultant of Parsing a Rule.

```
PERFORM until token is $ (end-of-rule)
    input a token
    IF the token is an operator
    THEN CASE:  token is
        "(":     push onto the temp stack the "(" and a pointer to the
                 current top of the operator stack (indicating the
                 beginning location of the operand to be defined);
        ")":     empty the temp stack onto the operator stack until ")"
                 is reached in the temp stack (the parentheses are not
                 pushed);
        "$":     empty the temp stack onto the operator stack;
        ".":     set a flag associated with the preceding operand
                 indicating that it has a name extension;
        "!":     push onto the operator stack the "!" and a pointer
                 into the operator stack where the last "(" occurred
                 (indicating the range of the operand);
        "]":     set a flag associated with the preceding operand
                 indicating that its value will have to be extracted
                 from the data base during execution of the rule;
        ",":     empty the temp stack onto the operator stack until "("
                 is reached in the temp stack (the "(" is not popped),
                 and push "," onto the operator stack;

        ELSE:    IF the previous token was not ( ) } ]̸ { [̸
                 THEN PERFORM until (temp stack is empty or the
                              precedence of the current token is >
                              the precedence of the operator at the top
                              of the temp stack)
                         CASE: precedence of current token compared to
                               precedence of the operator at the top of
                               the temp stack
                         =:    push onto the temp stack the token, its
                               precedence, and a pointer to the top of
                               the operator stack (indicating the
                               beginning of its "span of control" when
                               it is finally moved to the operator
                               stack);
                         >:    IF the preceding token was ")"
                               THEN push onto the operator stack the
                                    token and a pointer into the
                                    operator stack to the beginning of
                                    its operands (i.e., the operator
                                    stack length when the last "(" was
                                    encountered)
```

Figure 14. The PARSER Algorithm.

```
                               ELSE push onto the operator stack the
                                    token and a pointer to the
                                    preceding operator stack location
                                    (indicating that the previous
                                    location is its operand)
                               ENDIF:
                         <:    pop the temp stack, and push that data
                               (the operator and its operand pointer)
                               onto the operator stack;
                         ENDCASE
                 ENDIF
        ENDCASE
    ELSE CASE: token is
        object or relation: push the token onto the operand stack,
                 IF the previous token was not an operand
                 THEN a pointer to this token in the operand stack is
                      pushed onto the operator stack (indicating the
                      beginning of the operand string)
                 ENDIF;
        attribute: push the attribute onto the operand stack, set a
                 flag associated with that operand indicating that its
                 value will have to be extracted from the data base
                 during execution of the rule;
        literal: push the literal onto the operand stack, push a
                 pointer to the operand onto the operator stack;
        ENDCASE
    ENDIF
ENDPERFORM
```

Figure 14, Continued.

first occurrence, or a "next" occurrence if the string had been previously searched). If an occurrence is found, a TRUE is pushed onto a TRUE/FALSE (TF) stack, else a FALSE is pushed.

The routine is passed two parameters, the beginning and ending locations of the string in the operand stack. While finding an occurrence of an operand string, a trail of the data base keys of the discovered objects and relations is maintained. The keys are kept so that another occurrence of an operand string may be found in future calls to the FIND routine. Several strings may be stacked and "active" at any one time; the beginning stack location of the string serves as an identifier. This enables "next" occurrences of strings to be found. The search for a "first" occurrence of a string is forced by initializing the stack of keys for that string. Once an operand is found (or determined to be nonexistent), the operators may be easily applied.

The FIND operation gives special attention to name extensions (e.g., POSITION.2), assuring that the same or different data base keys are found as necessary. The ellipsis operator, !, requires additional attention to determine when additional iterations are needed, and if the iterations are entering an endless loop situation. The FIND algorithm is shown in Figure 15.

As an example for the execution of a rule, consider @ POSITION.1 MANAGES POSITION.2 Ɉ POSITION.1 EXECUTES ACTIVITY AUTHORIZED-BY POSITION.2 $. First, an occurrence of a position managing a different position is found. Then an occurrence of the second clause (where POSITION.1 and POSITION.2 are defined in the first clause) is searched

```
        IF the string has been previously successfully searched
        THEN pop the ellipsis goal last achieved (the ellipsis goal is
             the number of iterations used in the previous search if
             an ellipsis operator was involved),
             GO TO 4400 to find last object's next occurrence
        ENDIF
1120:   IF the operand appears previously in the rule
        THEN push the data base key already found,
             set a flag indicating a next occurrence of this operand
             is not allowed to be found
        ELSE find its first occurrence in the data base
             IF no occurrence exists
             THEN push an end-of-search flag,
                  PERFORM false-handler
             ELSE IF the object occurrence is required to be unique
                  THEN find next occurrence until it is unique
                  ENDIF
                  push the data base key of the found object
             ENDIF
        ENDIF
1990:   PERFORM pointer-handler
        IF the operand is an attribute
        THEN search for the attribute in the data base
             IF the attribute is found
             THEN fetch the attribute value
             ELSE IF a different occurrence of the object is allowed to
                     be found
                  THEN GO TO 4400 to find previous object's next
                       occurrence
                  ELSE PERFORM false-handler
                  ENDIF
             ENDIF
        ENDIF
        PERFORM pointer-handler
        IF the operand is a relation
        THEN search for the relation of the object in the data base
             IF the relation does not exist
             THEN IF a different occurrence of the previous object is
                     allowed to be found
                  THEN GO TO 4400 to find the next occurrence of the
                       previous object
                  ELSE PERFORM false-handler
             ENDIF
        ENDIF
        PERFORM pointer-handler
        IF the operand is a related object
```

Figure 15. The FIND Algorithm.

```
        THEN IF the relation can be made to a proper object occurrence
             THEN fetch the key of the nub,
                  handle the possibility of an infinite looping within
                  an ellipsis clause,
                  push the data base key of the nub
4400         ELSE IF backing up and finding a next occurrence of the
                     object is allowed
                  THEN find the next occurrence of the
                       previous object
                  ELSE PERFORM false-handler
                  ENDIF
             ENDIF
        ENDIF
        link up the object-relation-object loop such that the
        "related-to" object becomes the primary object,
        GO TO 1990
Pointer-Handler (determine which operand of the string to operate upon
next):
        IF at the end of a clause under the control of the ellipsis
           operator
        THEN increment the iteration counter
             IF the ellipsis operand has not iterated to its "goal"
             THEN reset the string pointer to the first operand of the
                  ellipsis operator
             ENDIF
        ELSE IF at the beginning of a clause under the control of the
                ellipsis operator
             THEN IF the ellipsis "goal" is zero
                  THEN set the string pointer to the operand beyond
                       the control of the ellipsis operator
                  ENDIF
             ENDIF
        ENDIF
        IF at the end of the string
        THEN push a TRUE onto the TF stack,
             push the current ellipsis goal
             RETURN
        ELSE increment the string pointer to the next operand
        ENDIF
```

Figure 15, Continued.

```
False-Handler (if an occurrence of the string cannot be found, perhaps
further iterations are needed of the ellipsis operator):
        IF an ellipsis operator is not in effect at the time of the
           failure to find an occurrence
        THEN increment the ellipsis goal,
             initialize the stack of keys,
             GO TO 1120 to begin the search anew
        ELSE push FALSE onto the TF stack,
             initialize the stack of keys,
             RETURN
        ENDIF
```

Figure 15, Continued.

for. If an occurrence is not found, execution continues; another occurrence of the first clause is found by looking for another POSITION.2. Control passes to the second clause again. Returning to the first clause, all POSITION.2's are found for a POSITION.1, and then the rule is repeated for all POSITION.1's. If (in this example) an occurrence of the second clause is found, then a violation report is printed.

All operators are concerned with the existence (or nonexistence) of operands (strings). Execution of a rule, therefore, involves invoking the FIND operation for particular operands. The FIND routine returns a logical true or false, indicating the success or failure of finding an occurrence of its operand. Basically, then, execution consists of finding operand occurrences and operating upon the returned true or false value. The differences among operators concern: 1) the location and number of operands in the rule, 2) whether additional occurrences of an operand are necessary, and 3) whether the desired result of a FIND is true or false.

If occurrences as specified by the rule cannot be found, then a "violation" of the rule is said to exist. When a violation is discovered, the values of the objects and attributes (if any) of the first clause are retrieved from the data base. The first clause and its values are printed out as a description of the violation. Execution of the rule then continues by searching for the next occurrence of the first clause.

The EXECUTE algorithm operates on each element of the operator stack, from bottom to top. If the stack element is an operand pointer, it is skipped; if the stack element is an operator, it is handled via a CASE construct. The algorithm makes heavy use of the FIND routine and a TRUE/FALSE (TF) stack. Values are pushed onto the TF stack by both the EXECUTE and FIND routines; elements are popped from the TF stack by the EXECUTE routine. Essentially, the remaining value of the TF stack after executing the rule expresses whether a violation to the rule has been found. The EXECUTE algorithm is shown in Figure 16.

```
PERFORM for each element of the operator stack
    IF the element is an operator
    THEN CASE: element is
        "@":    IF other operators are not involved in the first clause
                THEN FIND the first clause
                ENDIF
                IF POP(TF)
                THEN end of execution of this rule
                ELSE reset the ellipsis iteration counter (used in
                     FIND), reset data base keys list (used in FIND)
                     for all operands of second clause,
                ENDIF;
        "1":    IF other operators are not involved in the clause
                THEN FIND the clause
                ENDIF
                IF the "1" is in the first clause
                THEN reset the ellipsis iteration counter
                     reset data base keys list for all operands of the
                     second clause
                     IF LENGTH(TF) = 2
                     THEN IF BOTTOM(TF) = TRUE
                          THEN IF this is the first time for this
                               operator
                               THEN IF POP(TF) = TRUE
                                    THEN empty TF stack
                                    ELSE end of execution of rule
                                    ENDIF
                               ELSE report a violation,
                                    end of execution of rule
                               ENDIF
                          ELSE IF POP(TF) = TRUE
                               THEN empty TF stack
                               ELSE end of execution of rule
                               ENDIF
                          ENDIF
                     ELSE IF POP(TF) = FALSE
                          THEN there are no occurrences of the clause
                               in the data base,
                               end of execution of rule
                          ENDIF
                     ENDIF
                ELSE (the "1" is in the second clause)
                     IF POP(TF) = TRUE
                     THEN IF this is the second time for this operator
                          THEN PUSH(TF) = FALSE
```

Figure 16. The EXECUTE Algorithm.

```
                        ELSE set the operator stack pointer to the
                                beginning of the second clause
                        ENDIF
                ELSE IF this is the second time for this operator
                        THEN PUSH(TF) = TRUE
                        ELSE PUSH(TF) = FALSE
                        ENDIF
                ENDIF
        ENDIF
"⁆":    IF the rule has no first clause
        THEN FIND the second clause
        ENDIF
        set the operator stack pointer to the beginning of the
        rule
        IF POP(TF) = TRUE
        THEN PUSH(TF) = TRUE
        ELSE report a violation,
                PUSH(TF) = FALSE
        ENDIF;
"⁆̸":    (similar to "⁆");
"⁅":    set the operator stack pointer to the beginning of the
        rule,
        compare the values retrieved from the first clause to
        the operand values of the second clause
        IF any matches are found
        THEN PUSH(TF) = TRUE
        ELSE report a violation,
                PUSH(TF) = FALSE
        ENDIF;
"⁅̸":    (similar to "⁅");
"=":    PERFORM until something is pushed onto the TF stack
                FIND the first operand
                IF POP(TF) = FALSE
                THEN PUSH(TF) = FALSE
                ELSE FIND the second operand
                        IF POP(TF) = FALSE
                        THEN PUSH(TF) = FALSE
                        ELSE retrieve the values of the two operands
                                IF the values are equal
                                THEN PUSH(TF) = TRUE
                                ENDIF
                        ENDIF
                ENDIF
        ENDPERFORM;
"≠":    (similar to "=");
```

Figure 16. The EXECUTE Algorithm. -- Continued.

```
"S=":   PERFORM until all operand values are equal
            FIND an occurrence of the first operand
            IF POP(TF) = FALSE
            THEN PUSH(TF) = FALSE,
                BREAK
            ENDIF
            PERFORM for each other operand
                FIND an occurrence of the operand with a
                value equal to that of the first operand,
                IF an equal value cannot be found
                THEN BREAK
                ENDIF
            ENDPERFORM
            PUSH(TF) = TRUE
        ENDPERFORM
"S≠":   (similar to "S=");
"Λ":    IF the first operand is in fact an operator
        THEN (the FIND operation has already been performed)
        ELSE FIND the first operand
        ENDIF
        IF POP(TF) = FALSE
        THEN PUSH(TF) = FALSE
        ELSE IF the second operand is in fact an operator
            THEN (the FIND operation has already been
            performed)
            ELSE FIND the second operand
            ENDIF
            IF POP(TF) = TRUE
            THEN PUSH(TF) = TRUE
            ELSE PUSH(TF) = FALSE
            ENDIF
        ENDIF;
"V":    (similar to "Λ");
"SΛ":   PERFORM
            FIND an occurrence of the first operand
            IF POP(TF) = FALSE
            THEN PUSH(TF) = FALSE
            BREAK
            ENDIF
            PERFORM for each other operand
                FIND an occurrence of the operand
                IF POP(TF) = FALSE
                THEN BREAK
                ENDIF
```

Figure 16. The EXECUTE Algorithm. -- Continued.

```
                     ENDPERFORM
                     PUSH(TF) = TRUE
                ENDPERFORM;
        "SV":   (similar to "S∧");
        "!":    This operator is handled by the FIND routine.
                The EXECUTE routine performs only syntax checking.
        ENDCASE
    ENDIF
ENDPERFORM
```

Figure 16. The EXECUTE Algorithm. -- Continued.

CHAPTER 7

ACCEPTANCE AND EFFECTIVENESS OF THE METHODOLOGY

The preceding chapters have proposed a design of a computer aid for evaluating a plan of internal control. A language for documenting the plan of internal control, a language for expressing required and illegal constructs (rules) of the plan of internal control, and an algorithm for performing the evaluation by processing the rules against the documentation have been described. The manner in which the auditor would use the system has been discussed. Application of the system to a hypothetical case study is presented in the Appendix.

Studying the methodology's acceptance and measuring its effectiveness by an auditor in an actual audit have not been included in the scope of this dissertation. The methodology has been proven to work (that is, it is capable of finding weaknesses in a system of internal control) through its application to the hypothetical case study. Creating a methodology for an aid that is useful to internal control evaluation and is extendable to other application areas were the goals of this dissertation, and acceptance and effectiveness measurement are considered to be separate from these goals. This chapter ventures to discuss the expected degrees of acceptance and effectiveness of the automated evaluation system by an auditor in an actual audit situation. To gain a better appreciation of the expected acceptability, auditors have been interviewed to get their thoughts about the desirability and usefulness of the automated evaluation system. The findings of the interviews are summarized in Section 7.3.

Comparison of the nonautomated and the automated approaches to evaluating the plan of internal control serves as a starting point for the discussion of acceptance and effectiveness. The differences are summarized in Table 7, and a discussion of them follows.

7.1 Documentation

The manual procedure for evaluating the plan of internal control requires the auditor to record a description of the plan. This documentation may be in narrative form, flowchart form, or a combination of the two forms. The level of detail and the clarity of writing style varies among auditors; these are matters of personal preference or peculiarity. Often, an auditor relies on his or her memory for many of the details of existence and interrelationships of entities.

The automated procedure also requires documentation of the plan of internal control. This documentation, however, is very formal, and personal stylistics are eliminated. Further, automation requires that all relevant details be recorded, leaving nothing to the auditor's memory.

The acceptability of automation depends upon the auditor's desires. An auditor may insist on retaining the informality of manual procedures. In doing so, he or she is also opting to accept the higher probability of incomplete, inconsistent, and less comprehensible documentation. The formalization of the automated approach improves the

Table 7. Comparison of the Manual and Automated Approaches to Evaluation.

Manual	Automated
the plan of internal control might be flowcharted, or it might be documented in narrative form	the plan of internal control might be flowcharted, and it must be documented in the formal PSL/a language
different members of a team of documentors may use different writing styles and varying terminology	all members of the documentation team use the common, concise PSL/a language
documentation cost is a function of the time to create a narrative or flowchart description	documentation cost is a function of the time to create a flowchart description and convert it to PSL/a
the auditor may refer to lists for some evaluation criteria and rely on his or her memory for others	the auditor must explicitly state all evaluation criteria in the formal RULES language
processing time consists of the time required for the auditor to search for relations plus the time required to evaluate the relations after they are found	processing time consists of the time for the computer to find only those relations meeting specific constraints plus the time required for the auditor to evaluate the relation after it is found
it is difficult to find all relations of a particular entity in a narrative documentation	it is simple to find all relations of a particular entity in a network documentation
informality is possible, and documentation stylistics vary among documentors	rigid formality is necessary

completeness, consistency, and comprehensibility, which is expected to be desired by auditors.

Acceptance of the automated approach also depends upon its cost effectiveness. The audit cost is a function of time in terms of person-hours and computer-seconds. If the auditor were in the practice of flowcharting the plan of internal control under the manual procedure, then no additional time is required of the auditor under the automated approach, because the same flowchart is used. Converting the flowchart into PSL/a is a clerical task, the small cost of which should easily be offset by the gained benefits of completeness, consistency, and comprehensibility. If, under the manual procedure, the auditor writes the documentation in narrative form instead of flowchart form, then the tradeoff is in the time required to write narrative in the manual approach versus the time to write PSL/a in the automated approach; it is believed that the conciseness of PSL/a results in time savings.

A benefit that cannot be directly measured but that should affect the acceptance of the use of PSL/a is the standardization that the language provides. An audit usually consists of a team of auditors, each handling separate portions of the accounting system. PSL/a, as opposed to narrative or even flowcharting, encourages the use of common terminology and thereby links together the otherwise separate areas of documentation. This commonality later should lead to better comprehension and evaluation of the documentation.

Subsequent audits (the audit in the second year, or third year ...) are simplified by the formal documentation procedure. Whether the evaluation be manual or automated, the auditor must review and update the description of the system of internal control. Updating a narrative description is generally difficult and error prone. Updating a PSL/a documentation is simplified by the use of utility programs and made more accurate through automatic completeness and consistency checks.

7.2 Evaluation

To evaluate the plan of internal control, first the auditor must have an understanding of the plan through the provided documentation. While studying the descriptions, he or she must have in mind a set of valid rules or criteria by which to judge internal control. The rules must apply to whatever form the documentation takes. When evaluated "manually," he or she applies informal, unstated criteria that are based upon subjective feelings. While evaluating internal control, the auditor searches the questionnaire, narrative, or flowchart documentation for entities and relationships meeting or violating the criteria. This search is often laborious and error prone.

To automate the evaluation, the auditor must state the criteria in a formal language (RULES) and must be able to list all desired criteria. These rules are used to search the documentation, detecting violations of the criteria; the auditor still must pass judgment on the results of the search. Therefore, the methodology does not eliminate the evaluation process of the auditor. The methodology acts as an aid in applying the criteria by performing a faster and more reliable search of the documented system of internal control. Besides processing rules that have to search every entity and relation to report violations, the system may be used to report all characteristics of particular entities (a retrieval process). The auditor's questionnaires may be converted to RULES, and the rules may be applied interactively. It is a convenience to the auditor that all criteria do not have to be enumerated before beginning the evaluation process.

Over time, the set of rules for evaluation of a given client will grow. This is a benefit to the auditor in that, as the system of internal control changes, old rules may be easily reviewed, updated, and executed, and the auditor can give more concentration to the development of new rules. As the auditor gains experience and as new situations for testing are proposed, new rules will develop. All rules are transportable from one year to the next, and most are transportable from one client to the next. This transportability encourages completeness and consistency of evaluations over time and among cases.

7.3 Auditors' Evaluations of the Methodology

The preceding sections discussed my expectations of the acceptability of the methodology. To obtain other opinions, auditors were interviewed to assess their feelings on how acceptable, usable, and effective they expect the system to be in an actual audit situation. During the interview, the system was presented as a computerized tool to aid to whatever extent was desired the evaluation of the plan of internal control. The summary of the interviews presented in this section reports spontaneous comments by the auditors.

7.3.1 External Auditors' Evaluations

An external auditor's (Thierman, 1977) first impression was that the evaluation system would be very useful for the internal auditor -- more so than for the external auditor. The internal auditor could use it for planning the design of a system of internal control, studying the effect of adding or removing personnel and responsibilities. It could also be used for compliance testing by observing what functions employees are performing and comparing the findings to the PSL/a documentation.

This auditor also found many advantageous uses of the system for the external auditor. A problem that often occurs in documenting the plan of internal control is that it often is incomplete in that the documentor may believe certain aspects are obvious without documenting them; the automated system offers formalization that should lead to more complete documentation. Maintaining the documentation on the computer would allow updated flowcharts to be generated. Selective printouts of the documentation is very useful in situations such as the case in which the client thinks funds are being lost in the inventory area, and the auditor may request and then study only those controls related to inventory. For year-to-year auditing, the client may be allowed to keep and update a copy of the PSL/a documentation; at the time of the audit, the auditor may compare the client's version with the auditor's version from the previous year and concentrate on those areas found to differ. Clients tend to think that auditors merely perform tasks found on a checklist; computerization implies modernization, which is believed to impress clients, and the client would feel that a true evaluation is being performed instead of just the same chores that are performed every year. As a training tool, the automated system would be useful; the auditor trainee would be presented an internal control system, asked to document it in PSL/a (to test for completeness of work efforts), and asked to evaluate the controls based upon the documentation (to test the understanding of a documented system, and to test evaluation skills). Beyond training, a less experienced auditor would find the system useful by helping organize the documentation and evaluation functions.

This external auditor felt that the automated system would be used if no additional audit costs were incurred. If the costs were caused to increase, the system might still be used because of additional benefits gained. He realized that the availability of such a system would make available much information that was not originally requested but would be taken advantage of once it were made easily accessible.

Another external auditor (Jones, 1977) pointed out that the amount of reliance placed on internal controls varies among audit firms. Also, the amount of reliance placed on documentation varies among firms. This would directly affect the value of the automated system to an auditor.

On the positive side, this auditor felt the system would be good for bridging the areas of the documentation in the case of a large audit involving several documentors. As a planning tool, the system could help determine the effect of altering an old internal control system or of building a new one. The system would be more beneficial and acceptable if the audit firm made the automated system nationally available, supplying prepackaged sets of rules for different audit situations. Finally, this second auditor believed that an internal auditor would not use the system any differently than an external auditor would, although the internal auditor would generate a more detailed set of documentation.

7.3.2 Internal Auditor's Evaluation

An internal auditor's (Norrid, 1977) first impression was that the evaluation system would be better suited for an external auditor, and would be used only if audit costs would not increase. On the favorable side, he felt that the system would help generate more complete documentation. However, most of his impressions were not favorable.

This auditor attempted to foresee the system in actual use and questioned the ease with which the necessary details of internal control could be recorded. The desired level of detail would be difficult for the auditor to decide, because there is a point at which the greater level of detail requires too much documentation effort. He questioned whether someone would be willing to document to the level of detail such that one could determine, for example, if a transaction amount accounted for price markups and taxes.

Further, he does not like the formalization and regimentation that the system forces on the users. The internal audit is, in his case, an informal procedure, and this auditor wants the ability to evaluate accounting functions without using rigid guidelines.

This auditor was viewing the use of the system more for compliance and substantive testing than for evaluation of a plan of internal control, therefore foreseeing probable deficiencies in the system.

However, many of his comments are quite valid, considering that his desires (e.g., dislike of formalization) differ from those of some other auditors.

7.4 General Discussion

The automated system offers the auditor benefits beyond possible cost savings, work simplification, and completeness of work efforts. These benefits include improved communications, consistent application of terminology, consistent application of criteria over time, improved evaluation results, and expansion of criteria over time. Once the auditor automates the documentation aspect of the process, additional benefits may be realized in that the auditor learns that he or she is given a tool that offers the opportunity to improve the audit process, and the auditor may want to take advantage of the tool. The system is easily learned, but its acceptance will vary among users. This difference of opinion is based upon the amount of effort one is willing to expend on the required formalization, the desirability of formality, and the amount of emphasis that the auditor places on internal controls.

It is interesting to note that one external auditor believed that the system would be used better by an internal auditor, that the internal auditor believed that it would be used better by an external auditor, and that the second external auditor believed it would be equally useful for both types of auditors but that he would not like it for his own use. This type of reaction should be expected by the introduction of a new methodology and by a methodology that requires formalization of work methods.

Further benefits and uses of the methodology in auditing and other application areas are the subjects of Chapter 8.

CHAPTER 8

EXTENSIONS OF THE RESEARCH

A system for performing an automated evaluation of a plan of internal control has been designed and implemented. This chapter examines possible advances to the research and other applications of the research methodology. The chapter is concluded with a discussion of the feasibility of performing an entire financial audit by computer.

8.1 Extensions of the System's Components

The evaluation system consists of a documentation language and its associated data base maintenance routines, a "rules" language, and a software system for "executing" the rules. Later research efforts may add to the capabilities of these components.

8.1.1 Extensions to PSL/a

For analyses beyond the type currently available with the system, numerical information may need to be stored in the data base. For example, the triggering of an internal control action may depend upon the dollar amount of a transaction, or the study of the propagation of errors may require access to expected probabilities that control procedures will fail. These numbers may be stored through the definition of additional attribute-types, or online data files may be interfaced with the data base. PSL/a should be extended so that the modeler may identify whether a value is numeric or alphanumeric. These capabilities will not be difficult to implement, but were not chosen to be included in this prototype implementation.

8.1.2 Extensions to the RULES Language

A rule currently may include a test on the equality of two alphanumeric values. To extend RULES to allow for the comparison of numeric values, future versions of RULES should include arithmetic operators (+, -, *, /) and relative operators (/, >, ≥, <, ≤).

As an example shows, it is often convenient to allow the use of implied objects; since attribute-types and relation-types are unique to particular object-types, then all objects need not be explicity stated in a rule. For example, @ POSITION ∃ 1 POSITION MANAGED-BY $ should be allowed to be expressed as @ POSITION ∃ 1 MANAGED-BY $. Also, @ ACCOUNT [ACCOUNT-TYPE] ∈ ("ASSET", "LIABILITY", "EQUITY") $ should be allowed to be expressed as @ [ACCOUNT-TYPE] ∈ ("ASSET", "LIABILITY", "EQUITY") $. These capabilities will not be difficult to implement, but were not chosen to be included in this prototype implementation.

8.1.3 Valuation of Noncompliance with Rules

The current level of analysis performed by the evaluation system is the determination of compliance or noncompliance of the elements of the data base with the rules. One extension to the evaluation process is the study of the effect of noncompliance with a rule. For extended tests, it will be necessary for the auditor to assign values (or weights) to rules. The syntax for the rule will have to allow for the appended value (or values), and the evaluator software will have to appropriately apply the values.

One function of the values is to serve as weighting factors. The sum of the weights of violated rules may offer a measure of weakness of internal control (as discussed in Section 3.3.3). A rule could have an associated cost of a violation to the company and an associated cost of performing the internal control for performing a cost/benefit analysis (as discussed in Section 3.3.5).

To study the propagation of errors through the accounting system, expected probabilities of failure may be assigned to rules (as discussed in Section 3.3.5).

8.1.4 Data Dictionary

The greatest difficulty in building a model is the consistent naming of object values. For example, the process of informing customers of amounts due might be named "billing," "customer billing," "cust billing," "cust. billing," "A/R," or "accounts receivable." This problem is compounded when several persons are building the model.

Consistent naming is needed for two primary reasons. First, as the model is stored in the data base, two spellings of a value will result in two independent objects being stored. If BILLING GENERATES INVOICE and CUSTOMER RECEIVES CUST INVOICE, then INVOICE and CUST INVOICE are defined separately, and no flow can be traced between billing and the customer. Second, when defining rules, literals must appear as they do in the model. If a rule searches for the activity "accts receivable," it will not find the modeled activity "accounts receivable."

This problem is partially alleviated through the use of a data dictionary, and the use of one is suggested only as an aid to make the system easier to use. The data dictionary would contain all legal object-values and attribute-values. It would be referenced by the auditor for modeling and creating rules and by the data base loader and evaluator programs. The dictionary would also allow for the definition of synonyms. Any value not located in the dictionary would be flagged as an error.

It should be noted that LOADER.TAB currently used by the data base loader, and the synonym table currently used by the scanner, contain object-, attribute-, and relation-types, while the above discussion concerns object- and attribute-values. The loader table and synonym table could be incorporated into the data dictionary.

8.2 Extending to Further Auditing Functions

The research methodology has been applied to evaluating the plan of internal control. This section discusses applications of the methodology to other audit tasks.

8.2.1 Internal Control Compliance Testing

After evaluating the plan of internal control, audit practice requires that it be determined to what extent the plan has been implemented and is operational. This step, called compliance testing, is performed by observing and testing the actual flow of transactions through the processing and accounting systems.

The methodology may be applied to the plan of internal control, and then applied to compliance testing. In applying the methodology to compliance testing, a model would be built describing the internal control system as it is actually operating. The plan of internal control would comprise the rules, specifying what should exist in the implemented system of internal control. This would require transposing the format of the model of the original plan to that of the rules language.

8.2.2 Internal Auditing

Many of the functions of internal auditing overlap those of financial auditing (as discussed in Section 2.2). The internal auditor, concerned with the effectiveness of the system of internal control and efficiency of the general operation of the company, may make good use of the methodology for those overlapping functions and additional applications of similar nature.

8.3 Application to Nonauditing Functions

The developed system, considered from its physical design, consists of a language for specifying the contents of a data base of prescribed structure (objects, attributes of objects, and relations between objects), and a language and software system for searching the data base's contents. This section discusses applications of the methodology beyond that for the system of internal control.

8.3.1 Documentation

The modeling language and its associated data base maintenance software comprise in themselves a good documentation system (refer to Section 4.4.1). Creating and maintaining documentation manually is a difficult task; this is often the reason why documentation is poor.

Offering automated aids for documentation may encourage up-to-date and accurate documentation.

The system could be used to document internal control accounting systems, manufacturing process flow, systems design -- anything involving hierarchy, flow, or interrelationships. PSL is used as a documentation language, but its use is constrained to its prescribed set of objects, attributes, and relations. PSL/a has been designed as an extendable language, in that new types of objects, attributes, and relations may be added to fit the application area.

8.3.2 Evaluation of Other Data Base Models

The RULES subsystem may be used to determine completeness and consistency and to evaluate any of the models stored in the data base, as suggested in the previous section.

The constructs of the modeling and rules languages are fixed. However, their contents (object-types, attribute-types, and relation-types) are flexible and may be tailored to any application which fits in the general framework.

8.4 On Automation of the Entire Audit Process

This dissertation proposes a methodology for automating an evaluation of the plan of internal control. It begins with this step of the audit process because success was determined to be possible. After empirically proving the methodology to be feasible for the evaluation of the plan of internal control, the research could be extended to include additional processes of auditing. The goal of the overall research effort is to perform the entire audit process by computer.

Before taking such a bottom-up approach to slowly reaching the goal, consideration should be given first to the end goal to determine whether that goal is indeed obtainable. If those barriers that may prevent the achievement of the goal are identified, research efforts may proceed along the proper avenue. To identify those barriers and consider the feasibility of attaining the goal is the purpose of this section.

8.4.1 The Concept of Automation

In this presentation, a strong position is taken for a definition of automated auditing. To automate the audit process is to have a computer completely replace the auditor, except for the auditor's signature at the bottom of the audit report. It should be obvious that this goal is currently unachievable. However, such a strong definition offers a catalyst to provoke efforts of discovery of the types of obstacles that are to be found in trying to satisfy the goal. By identifying the obstacles, we can determine which aspects of the audit process may be automated currently and which areas require further research efforts.

The objective of the automated audit, then, is for the computer to evaluate enough evidence to offer support perhaps to the following version of the audit report (AICPA, 1976, §509.07):

> [Our computer has] examined the balance sheet of X Company as of December 31, 19XX, and the related statements of income, retained earnings and changes in financial position for the year then ended. [Its] examination was made in accordance with generally accepted [automated] auditing standards and, accordingly, included such tests of the accounting records and

> such other auditing procedures as [it] considered necessary in the circumstances.
>
> In our [computer's] opinion, the financial statements referred to above present fairly the financial position of X Company as of December 31, 19XX, and the results of its operations and the changes in its financial position for the year then ended, in conformity with generally accepted accounting principles applied on a basis consistent with that of the preceding year.

The scope paragraph states, among other things, that the computer followed generally accepted auditing standards. Do these standards allow for an automated audit? It also states that the computer performed such tests as it considered necessary. How will a computer be designed to "consider?"

The opinion paragraph states, among other things, that the client followed generally accepted accounting principles. The computer will have to determine adherence to generally accepted accounting standards, along with adherence to laws, codes, regulations, legal contracts, etc.

Application of the auditing standards is presented in the next section. This is followed by a discussion of the informational content of the audit process, pointing out the obstacles that may hamper the feasibility of complete automation.

8.4.2 Applying the Auditing Standards to a Computerized Audit

Consideration is given to applying the ten generally accepted auditing standards (AICPA, 1976, §150.02) to an audit that is to be performed completely by a computer -- instead of by a human.

General Standards.

"1. The examination is to be performed by a person or persons having adequate technical training and proficiency as an auditor" (AICPA, 1976, §150.02). Obviously intended for the human auditor, the mechanized auditor will express training and proficiency through the quality of its software and input data. It will be necessary for the human auditor to participate in the design and control of the automated auditor.

"2. In all matters relating to the assignment, an independence in mental attitude is to be maintained by the auditor or auditors" (AICPA, 1976, §150.02). The intent is to remove bias due to conflicts of interests while performing the audit. Here, the independence is reflected in the design and implementation procedures of the automated system.

"3. Due professional care is to be exercised in the performance of the examination and the preparation of the report" (AICPA, 1976, §150.02). If the performance is by machine, the "due professional care" must be redefined to take into consideration how the machine is instructed to perform (i.e., how the audit function is programmed).

Standards of Field Work.

"1. The work is to be adequately planned and assistants, if any, are to be properly supervised" (AICPA, 1976, §150.02). Planning and supervision are built into the system. Once automated auditing has begun, human intervention is not allowed. Therefore, the automated auditor must anticipate all possible circumstances.

"2. There is to be a proper study and evaluation of the existing internal control as a basis for reliance thereon and for the determination of the resultant extent of the tests to which auditing procedures are to be restricted" (AICPA, 1976, §150.02). The purpose of this standard is to reduce the work of compliance and substantive testing of transactions and balances to a minimum by determining a measure of reliance which may be placed on the manner in which the accounting information is generated. A computer, though, does not tire, so perhaps consideration should be given to replacing the evaluation of the internal controls by complete recomputation of accounting data. This is further discussed at a later point.

"3. Sufficient competent evidential matter is to be obtained through inspection, observation, inquiries, and confirmations to afford a reasonable basis for an opinion regarding the financial statements under examination" (AICPA, 1976, §150.02). Sufficiency and competency must be quantified to be evaluated by computer. How does a computer inspect, observe, inquire, and confirm? These procedures are designed for human processing; a different approach is required for computer processing. A problem which arises is the specification and collection of the input data. This is further discussed at a later point.

Standards of Reporting.

"1. The report shall state whether the financial statements are presented in accordance with generally accepted accounting principles" (AICPA, 1976, §150.02). Generally accepted accounting principles will need to be made not only computer-accessible but also computer-interpretable.

"2. The report shall state whether such principles have been consistently observed in the current period in relation to the preceding period" (AICPA, 1976, §150.02). Access to the current and preceding years' data must be possible. A manner must be developed to determine how the principles were applied and how to evaluate the exceptions.

"3. Informative disclosures in the financial statements are to be regarded as reasonably adequate unless otherwise stated in the report" (AICPA, 1976, §150.02). All types of disclosures which could be appended to the statements would need to be enumerated so that the computer may determine those that may be applicable. A method of objectively measuring adequacy must be developed.

"4. The report shall either contain an expression of opinion regarding the financial statements, taken as a whole, or an assertion to the effect that an opinion cannot be expressed. When an overall opinion cannot be expressed, the reasons therefore should be stated. In all cases where an auditor's name is associated with financial statements, the report should contain a clear-cut indication of the character of the auditor's examination, if any, and the degree of responsibility he is taking" (AICPA, 1976, §150.02). This standard requires the weighing of all evidence obtained in the audit process and again involves problems of subjectivity.

Many questions have been raised concerning the applicability of generally accepted auditing standards and procedures. It must be realized that the standards and procedures have been designed for a manual, human auditor.

Automation of the audit process may require an entirely different approach than that of manual auditing. The automation attempt cannot be dismissed on this basis, as new standards can evolve. This discussion of auditing standards, alone, emphasizes the need for research into the audit process itself, identifying its underlying principles and determining new effective approaches to achieving its objective. The primary aspect of the audit that carries through to the automated audit is the audit objective: to determine the fairness of the statements' presentations.

8.4.3 Enumerating, Collecting, and Accessing Audit Information

An approach to discovering the obstacles which should be expected in the attempt to automate the audit is to consider the informational content of the audit process. The financial audit can be considered as a process having two major sets of inputs and one output (the audit report).

The auditor brings one set of inputs to the audit process. These inputs are the evaluation techniques and criteria, consisting of auditing standards, generally accepted accounting principles, guidelines, plans, questionnaires, weighting factors, concepts of materiality, evaluation methods, prior audit information, experience, and creativity.

Also included in the information set supplied by the auditor is information describing requirements and restrictions that have been placed on the client by external sources. The requirements are found in laws, regulations, guidelines, rules, codes, contracts, and other legal documents.

The other major set of inputs is the information concerning the client that is to be audited, including a description of the economy, the industry, the client in general, its operating procedures, transactions, personnel, organizational charts, chart of accounts, accounting system, managerial goals, accounting records, documents, contracts, leases, deeds, inventory, tangible and intangible property.

The first problem that must be solved is the enumerating of the elements of the input sets. All forms of information that could be necessary for the audit of a particular client need to be identified. The preceding paragraphs list several of these information sources, but the list is certainly incomplete. An incomplete list of sources may lead to an oversight in the audit process. An additional problem, therefore, is the determination of completeness of the information sets. Solutions to these problems are not offered in this presentation, but rather the point is made that these are obstacles to be overcome.

Once it is determined what information is needed, two problems immediately arise. First: How is the information obtained? Second: How is it made computer-accessible? These two questions are considered simultaneously as they are applied to a sampling of the inputs.

Considered first is the representation of data describing the client (that information that is to be audited).

It has already been shown (Lieberman, 1977) that the plan of internal control can be formally modeled. The implemented system of internal control can also be modeled for compliance and substantive testing.

Can something so ill-defined as a description of the economy be modeled? Those aspects that are of interest would have to be defined, but "interest" is subjective, and the economy never being completely stable always allows the possibility of something new being of interest.

Contracts and other documents can be easily put in a computer-accessible form, but the auditing computer must state to which documents it wants access. Unless the computer is allowed to make the blanket request, "GIVE ME ALL DOCUMENTS," the list of expected documents or criteria for selecting desired documents must be enumerated. Yet, the enumeration is identical to the paradox of determining completeness of a model: It cannot be determined that all desired entities have been modeled unless a set of all desired entities is provided that enumerates all entities that should be modeled (itself being a model)!

Interviews with personnel are major sources of information. The difficulties involved in a human-to-computer interface are obvious and will not be pursued. Assuming that the computer has an audio input device, and assuming an English semantic analyzer is developed, and even assuming that a set of questions has been enumerated, there are always hidden thoughts in any verbal answer. An auditor recognizes a hesitation, a shift downward of the eyes, words chosen to imply a second meaning. Can a computer be made to recognize these, and then determine how to coax the interviewee into verbalizing those held-back thoughts?

Consideration is now given to those inputs with which the auditor is equipped at the onset of the audit process.

The set of all applicable laws, regulations, principles, techniques, and criteria can be enumerated, though lengthy it may be. The information can even be made computer-accessible. To make it computer-interpretable is another consideration. Many interpretations can be made of a law. They are subject to precedents, special situations, and subjective evaluation. Subject evaluations may vary among auditors, making the definition of one common algorithm infeasible.

An auditor gains experience over time. Experience leads to the development of new auditing techniques (additional inputs) after the recognition of errors due to previously inadequate audit practices. The process of experience, or learning, can be programmed (e.g., via game theory) to the extent that errors or inadequacies can be recognized a second time and that some corrective action is to be triggered. Recognizing errors or inadequacies in audit procedures the first time the errors occur is one problem to be overcome. Determining the proper corrective action is another problem to be overcome, because unless a set of all possible corrective actions is enumerated, creativity is a requisite.

Creativity, the generation of new types of inputs (perhaps techniques for testing accounting procedures), is a major aspect of auditing. The audit is not at this point in time a deterministic process involving a prescribed algorithm leading to the final opinion. Instead, the auditor considers the circumstances of all the inputs and creates a plan. During the process, the auditor recognizes new circumstances and creates alternative plans. The current state of computer technology does not allow for the creation of alternative actions.

8.4.4 Summary of Problems and Possible Solutions

Enumeration of the types of information needed for the audit process is possible; this lengthy list would most likely be developed over time through several audit experiences. The greater difficulty is the enumeration of the possible occurrence of each type of information. For example, it can be stated that property deeds are required, but it cannot be determined that all property deeds have been made available. People have the ability to hide information from the auditor and certainly from the auditing computer, and this obstacle is not computer-solvable.

Much information has not been quantified; instead it is left to the discretion of the auditor to apply subjective evaluation. Examples are conditions of the economy and levels of materiality. It may be possible to quantify this information, but only after the auditing profession agrees to these values. Agreement among auditors on the quantification of currently subjective measurement is improbable, because the subjective determination of materiality levels (for example) has many ill-defined parameters, and each audit engagement is a unique circumstance.

Interfacing the human with the computer is one of the greatest obstacles to overcome. Systems have been developed for understanding limited vocabularies, but there is much to be developed before the human can freely converse with the machine. The machine-interpretation of spoken English is possible and will become a reality in the future. Human interfacing also involves the discovery of hidden meaning in a sentence and other similar problems.

If the audit is to be performed by computer, then there is one case that alleviates the difficulties involved in the representation of the client data that are to be audited. If the original source of all accounting information is maintained on and generated by a computer, then computer-accessibility is no obstacle. The human would have to be completely removed from accounting transactions. For example, when a customer purchases a product the customer interacts with a machine that in turn counts cash or accesses the computer of the customer's bank; when the corporation obtains property the exchange is performed by machines and the documents are maintained by machines. By removing the human, much of the information previously declared as being required to perform the audit is no longer needed; much information is obtained in the manual audit as a source for the determination of motives for people to commit fraud or catalysts for errors. A computer, being a deterministic machine, cannot motivate or prompt itself to err.

Shifting the responsibility of processing from people to machines incurs a further problem. The audit computer must be able to analyze the processing method of the client computer. The analysis of software, vis-à-vis "program proving," has not been proven possible, though. This certainly has a better chance of success than that of the audit computer trying to analyze the ambiguous and ill-defined processing methods of people.

The final area of obstacles to be considered involves the information set that guides the audit process. The information set includes requirements of the accounting data (laws, regulations, contracts, etc.) and auditing techniques and procedures. The problem of enumeration was considered and determined to be feasible up to the state of creativity. Unless the input sets can be exhaustively enumerated, the human auditor's ability of creativity cannot be replaced.

A possible solution evolves from the proposal of quantifying the audit process. If all procedures and weighting factors and evaluations are quantified, then creativity is eliminated. Quantification is feasible under the assumption of a completely automated accounting system; in this case the audit is simply a matter of complete recomputation.

Such a step would require redefinition of generally accepted auditing standards. These standards are designed for a human auditor and do not necessarily apply when the auditor is a machine, as was earlier discussed. For example, the second standard of field work requires a study and evaluation of the system of internal control so that the auditor may determine how much (if any) reliance to place on it. The auditor wants to rely on the system so that the audit effort will be lessened. Even if no internal control system exists, the auditor may still proceed with the audit; the auditor usually does not proceed because of the extensive substantive testing that would be required.

Replacing the auditor with a nontiring machine could eliminate the need to evaluate the system of internal control; let it assume no control exists and proceed with lengthy substantive testing.

8.4.5 Conclusions Concerning Automated Auditing

Complete automation of the financial audit process was never thought to be a feasibility, but by considering it we have gained insight into the audit process and discovered areas of needed research. A major advancement in computer technology is required; the human-to-computer interface will be very difficult to complete; and the evaluation of accounting information is too subjective with little quantifiable basis. Much support has been given in this presentation for the need to evaluate the audit perspective and to redesign the audit standards and procedures to allow for more effective auditing in a new era. Complete automation is a goal for which to aim, and, within the foreseeable future, techniques that are developed toward that goal will be recognized as beneficial aids to the audit process.

APPENDIX

CASE STUDY

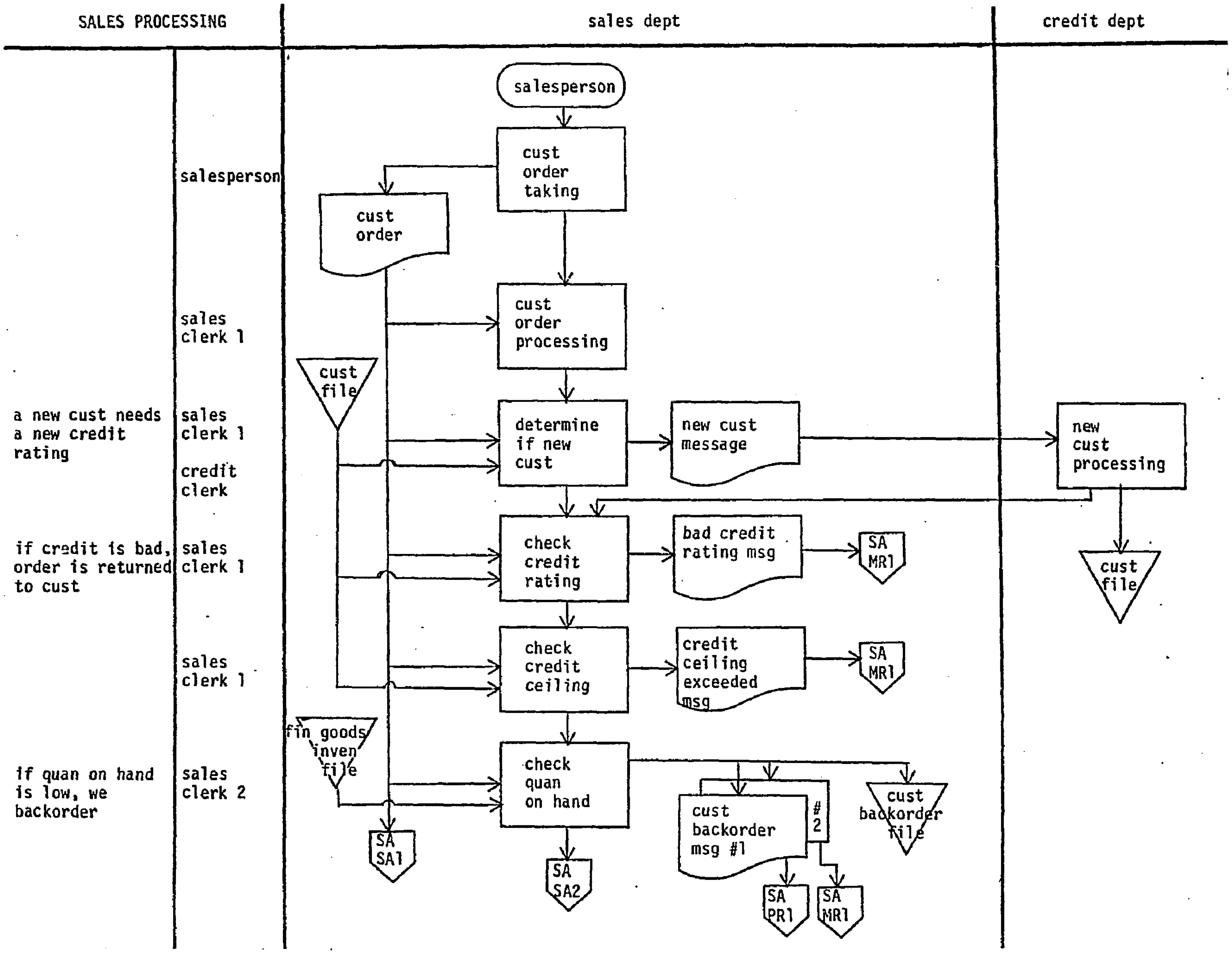
A test case for the research methodology has been applied to "Company Z" (Nunamaker and Konsynski, 1975), itself a test case for the design of information systems. The Company Z description was lacking a substantial accounting system; it was, therefore, modified and extended to be suitable for a study of internal control.

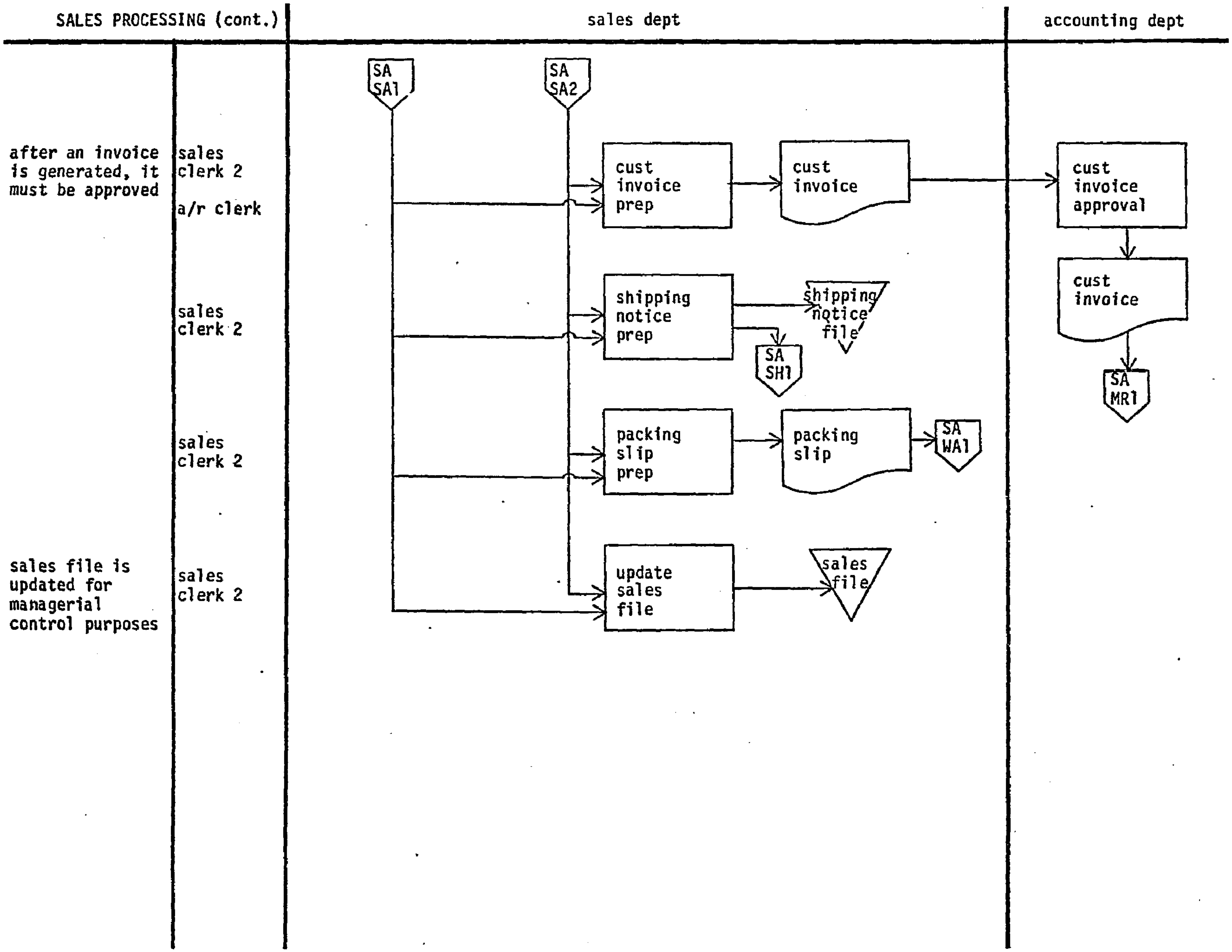
It is suggested in Section 4.2.1 that the description of the system of internal control be flowcharted before creating the PSL/a description. Company Z was made applicable as a test case by extending its flowchart description; a written description of Company Z's internal control system, therefore, is not presented.

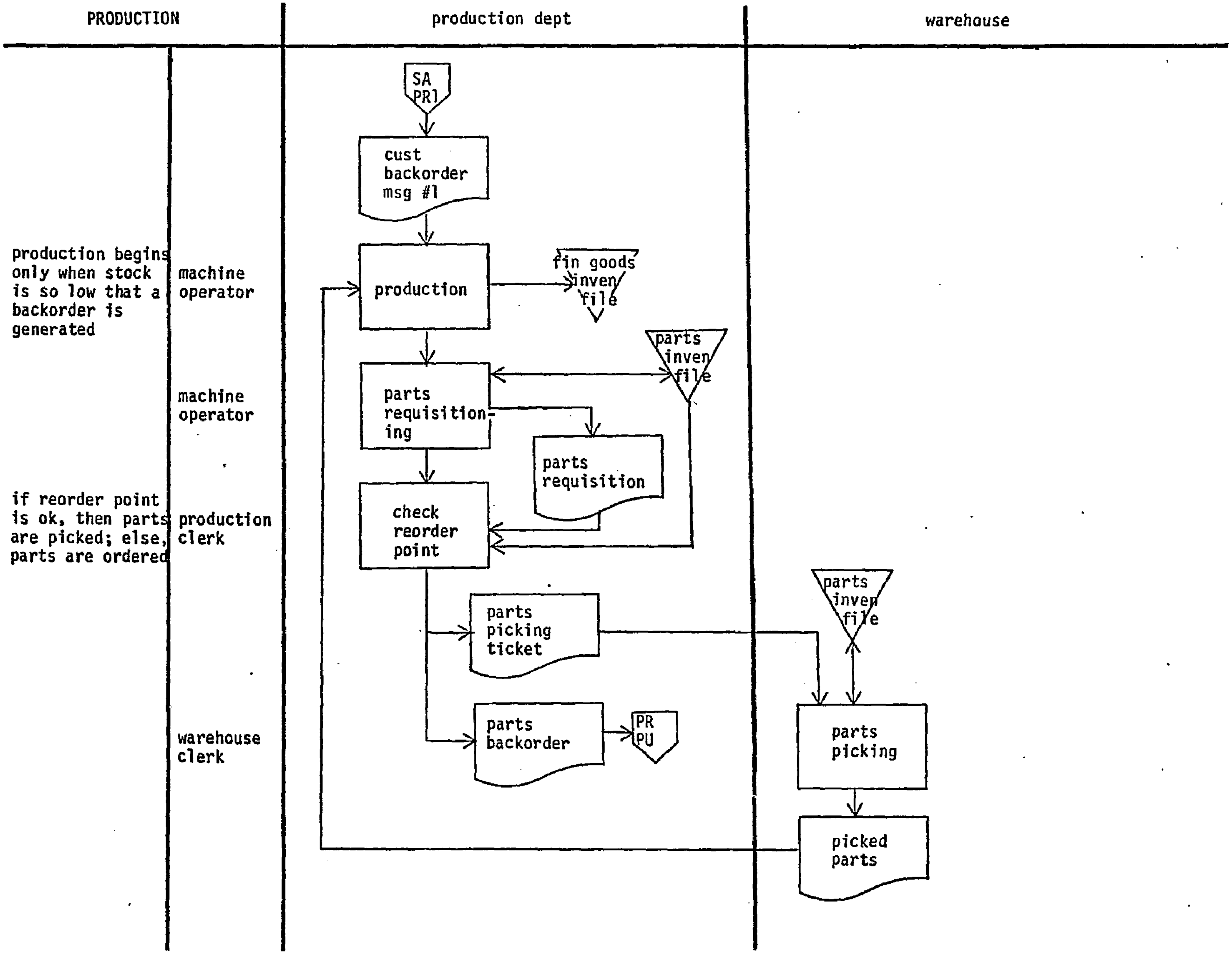
As an initial research trial, it was not considered appropriate to select an actual company as a test case. A major effort of the research was the determination of just what was desired of the research. Initially, it was not known what was to be modeled and in what form it would appear. Nor was it completely clear how the rules were to be applied. A fabricated case study allows for patience and offers the flexibility required of initial research.

The presentation of the case study illustrates all the steps involved in documenting and evaluating the plan of internal control. The reader is led through the user/computer interactions. The interactions are accompanied by narratives explaining the inputs and outputs.

The following aspects of Company Z were modeled: sales processing, production, vendor payments, shipping, customer payments, miscellaneous reporting, and miscellaneous activities. The flowcharts for these areas appear on the following pages.

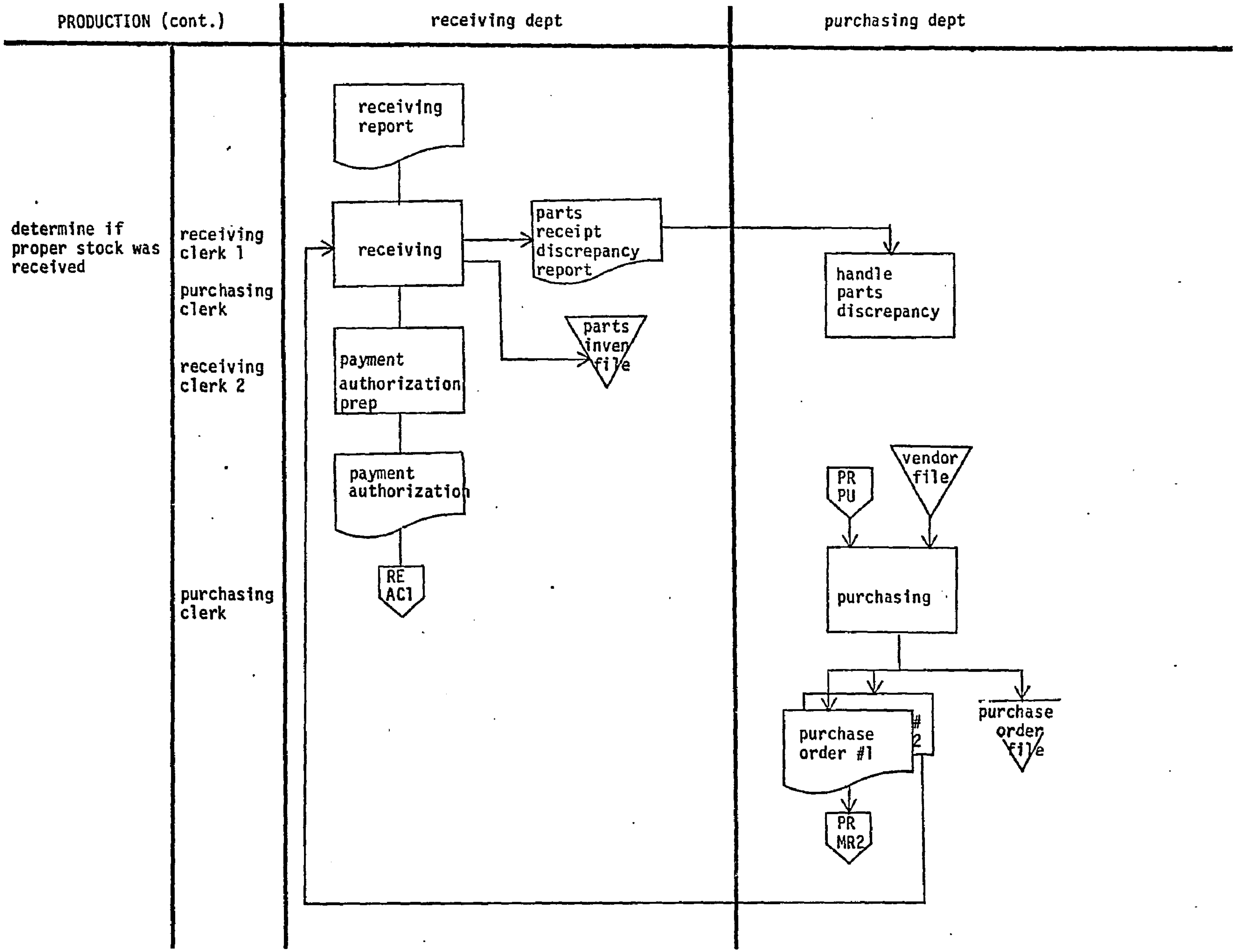

SALES PROCESSING
sales dept
credit dept
salesperson
salesperson
cust order taking
cust order
sales clerk 1
cust order processing
cust file
a new cust needs a new credit rating
sales clerk 1
credit clerk
determine if new cust
new cust message
new cust processing
cust file
if credit is bad, order is returned to cust
sales clerk 1
check credit rating
bad credit rating msg
SA MR1
sales clerk 1
check credit ceiling
credit ceiling exceeded msg
SA MR1
fin goods inven file
if quan on hand is low, we backorder
sales clerk 2
check quan on hand
cust backorder msg #1
#2
cust backorder file
SA SA1
SA SA2
SA PR1
SA MR1


SALES PROCESSING (cont.)
sales dept
accounting dept
SA SA1
SA SA2
after an invoice is generated, it must be approved
sales clerk 2
a/r clerk
cust invoice prep
cust invoice
cust invoice approval
cust invoice
SA MR1
sales clerk 2
shipping notice prep
shipping notice file
SA SH1
sales clerk 2
packing slip prep
packing slip
SA WA1
sales file is updated for managerial control purposes
sales clerk 2
update sales file
sales file



production begins only when stock is so low that a backorder is generated

if reorder point is ok, then parts are picked; else, parts are ordered


PRODUCTION (cont.)
receiving dept
purchasing dept
determine if proper stock was received
receiving clerk 1
purchasing clerk
receiving clerk 2
purchasing clerk
receiving report
receiving
parts receipt discrepancy report
parts inven file
payment authorization prep
payment authorizatioh
RE AC1
handle parts discrepancy
PR PU
vendor file
purchasing
purchase order #1
#2
purchase order file
PR MR2

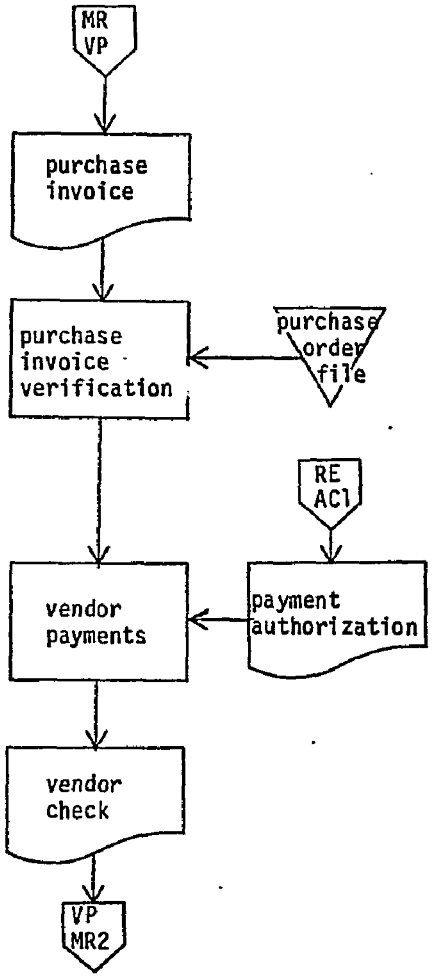
VENDOR PAYMENTS

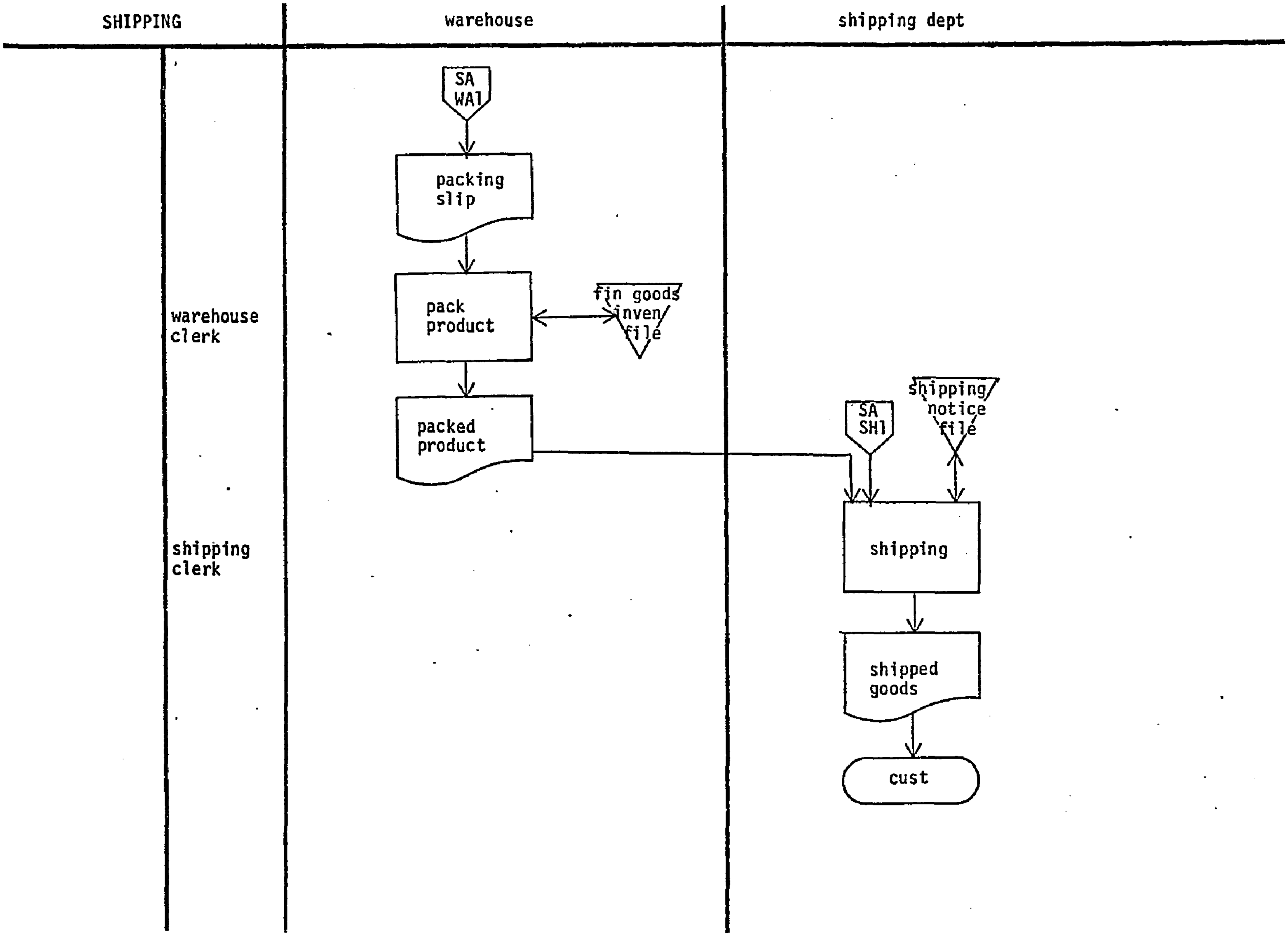
accounting dept

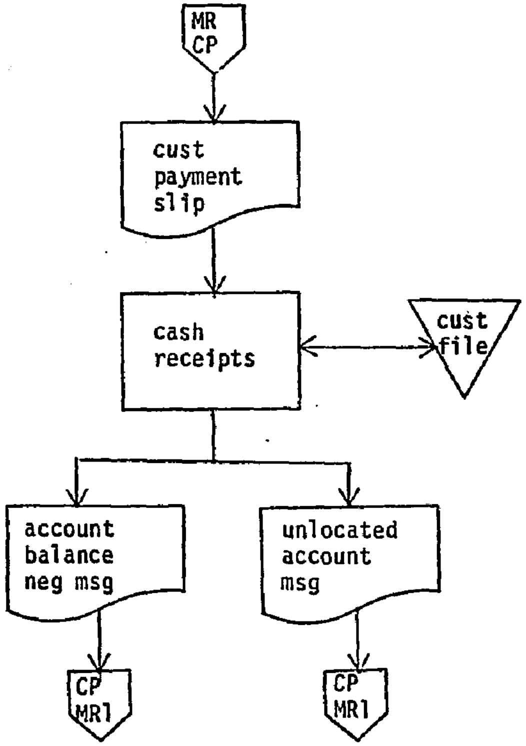
compare what we
ordered to what
vendor billed

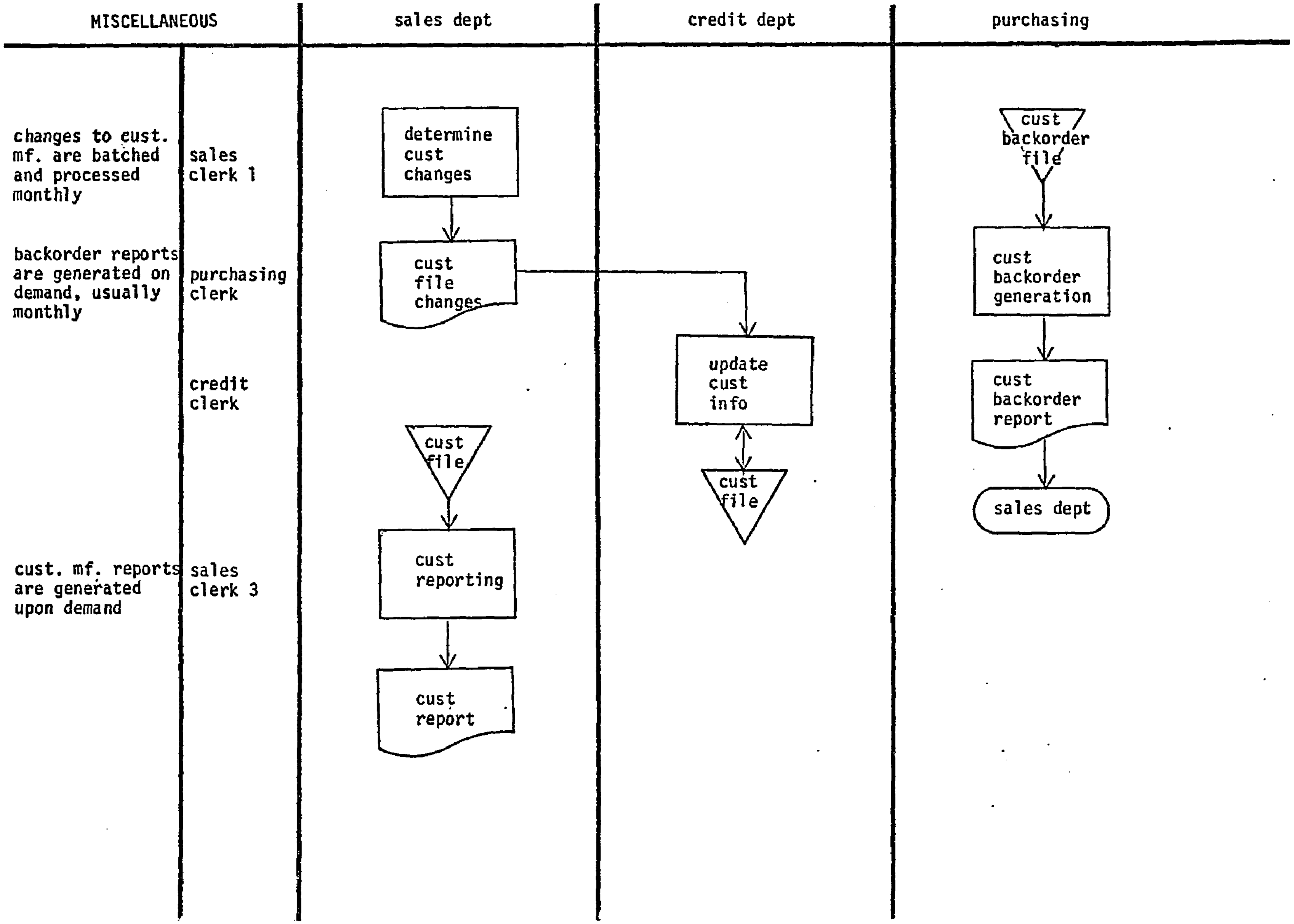
a/p clerk

a/p clerk




SHIPPING
warehouse
shipping dept
warehouse clerk
shipping clerk
SA WA1
packing slip
pack product
fin goods inven file
packed product
SA SH1
shipping notice file
shipping
shipped goods
cust

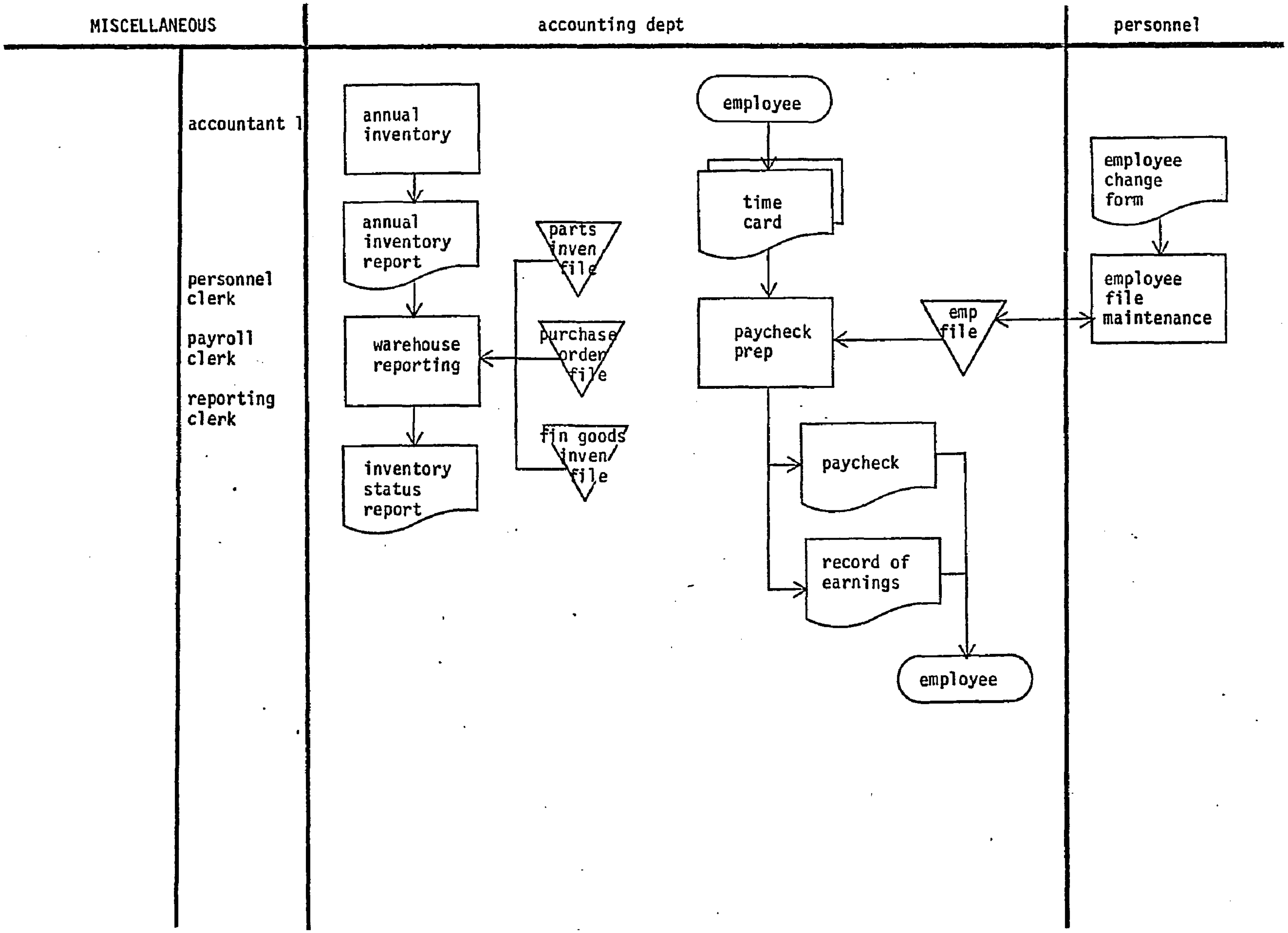
CUSTOMER PAYMENTS		accounting dept
upon receipt of payment, either update the cust. account or print an error message	a/r clerk	

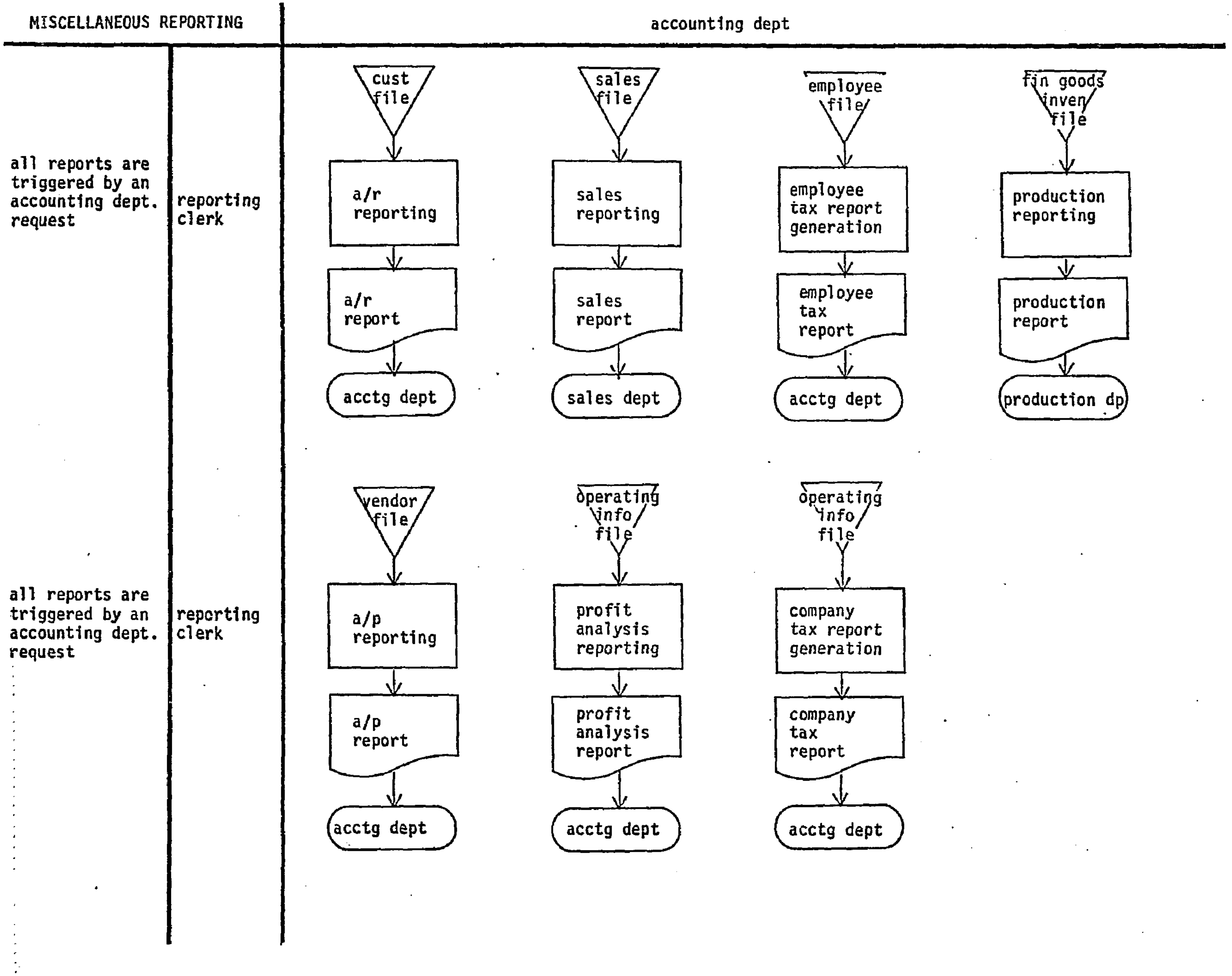


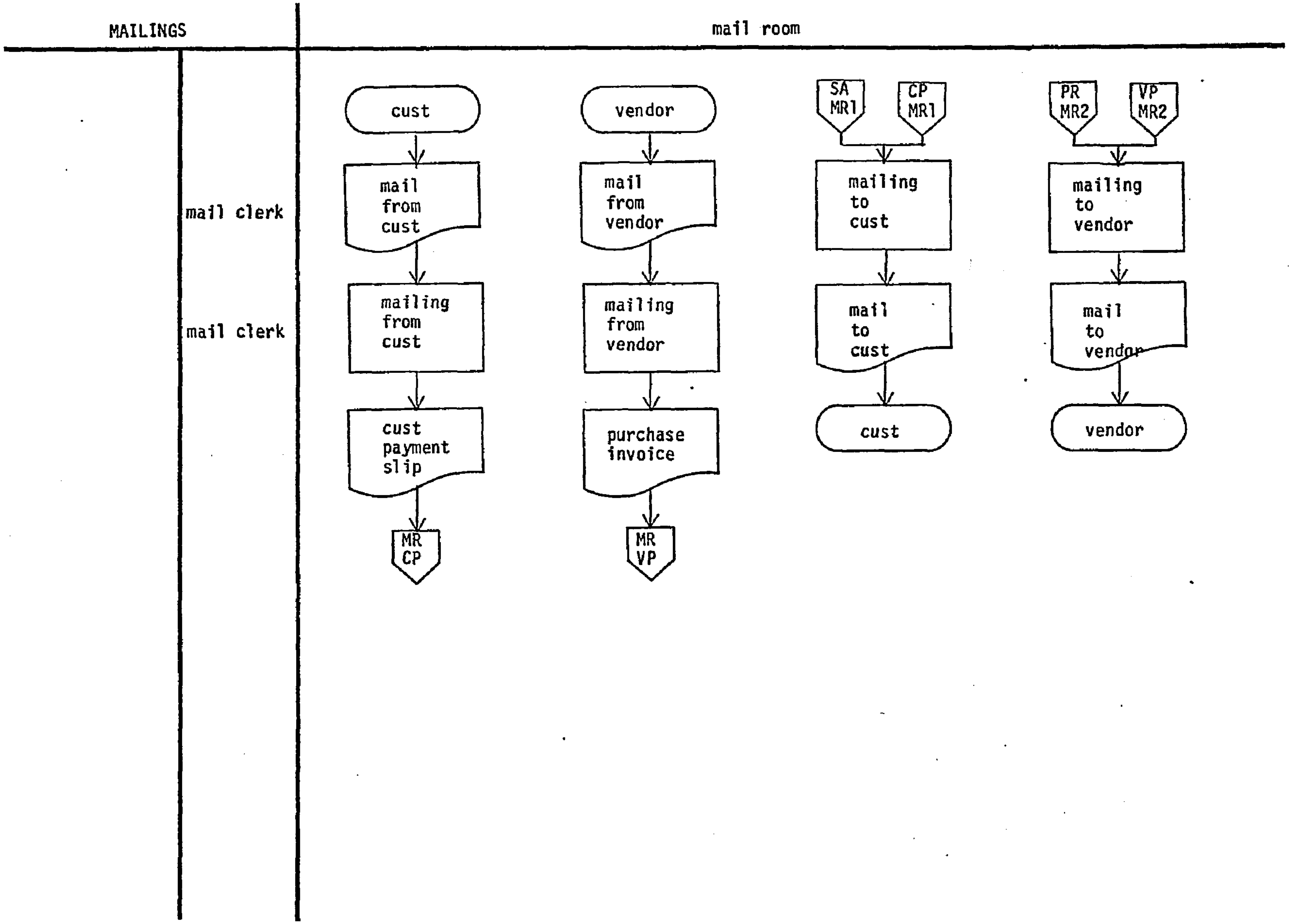
changes to cust. mf. are batched and processed monthly

backorder reports are generated on demand, usually monthly

cust. mf. reports are generated upon demand


MISCELLANEOUS
accounting dept
personnel
accountant 1
personnel clerk
payroll clerk
reporting clerk
annual inventory
annual inventory report
warehouse reporting
inventory status report
parts inven file
purchase order file
fin goods inven file
employee
time card
paycheck prep
emp file
paycheck
record of earnings
employee
employee change form
employee file maintenance


MISCELLANEOUS REPORTING
accounting dept
all reports are triggered by an accounting dept. request
reporting clerk
cust file
a/r reporting
a/r report
acctg dept
sales file
sales reporting
sales report
sales dept
employee file
employee tax report generation
employee tax report
acctg dept
fin goods inven file
production reporting
production report
production dp
all reports are triggered by an accounting dept. request
reporting clerk
vendor file
a/p reporting
a/p report
acctg dept
operating info file
profit analysis reporting
profit analysis report
acctg dept
operating info file
company tax report generation
company tax report
acctg dept


MAILINGS
mail room
mail clerk
mail clerk
cust
mail from cust
mailing from cust
cust payment slip
MR CP
vendor
mail from vendor
mailing from vendor
purchase invoice
MR VP
SA MR1
CP MR1
mailing to cust
mail to cust
cust
PR MR2
VP MR2
mailing to vendor
mail to vendor
vendor

All of the system's programs guide the user with prompts, requesting file names, commands, and other parameters.

On the following page are the computer interactions necessary to initialize the data base. The DDLA program builds a set of tables describing the data base schema. Its input file is MOF.DDL, which appears following the computer interactions. The TEMP.DAT file lists the amount of storage required by the schema tables.

The DBIN program creates an empty data base of ten pages. The data base management system automatically creates additional empty pages as needed.

The tables and initial data base require 43 blocks of storage. After the data base is totally loaded, it will use 35 pages (of which 5 are empty) occupying 142 blocks of storage.

```
LOGIN 3305/10555

U of Ariz  6.02 KL  2:21 P.M. Thur Feb 17
Job 82  TTY230  User ARTIE
Balance = $205.02

.RUN STDB:DDLA[2040,10103]

ENTER DDL INPUT FILE SPECIFICATION
MOF.DDL
ENTER DATA BASE NAME
Z
ENTER DDL ANALYZER REPORT FILE SPECIFICATION
DDL.RPT
ENTER BLOCK DATA REPORT FILE SPECIFICATION
TEMP.DAT
STOP

END OF EXECUTION
CPU TIME: 0.79  ELAPSED TIME: 1:18.32
EXIT

.RUN STDB:DBIN[2040,10103]

ENTER DATA BASE NAME
Z
ENTER DBIN REPORT FILE SPECIFICATION
TTY:
 DATA BASE INITIALIZED WITH    10 PAGES.

END OF EXECUTION
CPU TIME: 0.78  ELAPSED TIME: 39.03
EXIT

.PRINT/FORMS:WN/DEL/FI:FO DDL.RPT,TEMP.DAT
Total of 16 blocks in 2 files in LPT request

.K/F
Job 82, User [3305,10555] Logged off TTY230   1428  17-Feb-77
Runtime 2.60 Sec; Session Charge $ 0.38
```

```
*       DDL DESCRIBING THE DATA BASE SCHEMA

NPAGES  10

RECORD  OBJECT
ITEM    OBTYPE  CHAR    20
ITEM    OBNAME  CHAR    20

RECORD  ATTRIB
ITEM    ATTYPE  CHAR    20
ITEM    ATVALU  CHAR    20

RECORD  RELATE
ITEM    RETYPE  CHAR    20

RECORD  NUB


SET     SYSOBJ  SORTED  OBTYPE
OWNER   SYSTEM
MEMBER  OBJECT

SET     OBJNAM  SORTED  OBNAME
OWNER   SYSTEM
MEMBER  OBJECT

SET     OBJREL  SORTED  RETYPE
OWNER   OBJECT
MEMBER  RELATE

SET     RELNUB  IMMAT
OWNER   RELATE
MEMBER  NUB

SET     OBJNUB  IMMAT
OWNER   OBJECT
MEMBER  NUB

SET     OBJATT  SORTED  ATTYPE
OWNER   OBJECT
MEMBER  ATTRIB
```

A sample of the data for loading the PSL/a data base follows. The entire set is in two parts. The first part is the model as originally created from the Company Z flowcharts. That data set was subsequently loaded into the data base, and completeness testing was performed on it. An additional set of model data was created and loaded into the data base. The loading (refer to Section 4.3.2) of the two data sets and testing of the first data set are not illustrated here. Subsequent testing and loading are later illustrated.

```
ACTIVITY             MAILING TO VENDOR
EXECUTED-BY          MAIL CLERK
STIMULATED           INTERNALLY
GENERATES            MAIL TO VENDOR
EXTRACTS             VENDOR CHECK
EXTRACTS             PURCHASE ORDER #1

ACTIVITY             MAILING TO CUST
EXECUTED-BY          MAIL CLERK
STIMULATED           INTERNALLY
GENERATES            MAIL TO CUST
EXTRACTS             BAD CREDIT RATING MSG
EXTRACTS             CREDIT CEILING EXCEEDED MSG
EXTRACTS             CUST INVOICE
EXTRACTS             UNLOCATED ACCOUNT MSG

REPORT               MAIL TO VENDOR
RECEIVED-BY          VENDOR

REPORT               MAIL TO CUST
RECEIVED-BY          CUST

ACTIVITY             MAILING FROM CUST
EXECUTED-BY          MAIL CLERK
STIMULATED           EXTERNALLY
GENERATES            CUST PAYMENT SLIP

ACTIVITY             MAILING FROM VENDOR
EXECUTED-BY          MAIL CLERK
STIMULATED           EXTERNALLY

REPORT               MAIL FROM CUST
PREPARED-BY          CUST

REPORT               MAIL FROM VENDOR
PREPARED-BY          VENDOR

        ***** CUST ORDER *****

ACTIVITY             CUST ORDER TAKING
EXECUTED-BY          SALESPERSON
STIMULATED           EXTERNALLY
PERFORMED-BY         SALES DEPT
TRIGGERS             CUST ORDER PROCESSING
GENERATES            CUST ORDER

REPORT               CUST ORDER
GENERATED-BY         CUST ORDER TAKING
RECEIVED-BY          SALES DEPT
EXTRACTED-BY         CUST ORDER PROCESSING

ACTIVITY             CUST ORDER PROCESSING
EXECUTED-BY          SALES CLERK 1
STIMULATED           INTERNALLY
PROCESS              VERIFY ACCURACY & COMPLETENESS
PERFORMED-BY         SALES DEPT
EXTRACTS             CUST ORDER
TRIGGERS             DETERMINE IF NEW CUST

ACTIVITY             DETERMINE IF NEW CUST
EXECUTED-BY          SALES CLERK 1
```

```
STIMULATED          INTERNALLY
PERFORMED-BY        SALES DEPT
GENERATES           NEW CUST MSG
TRIGGERS            NEW CUST PROCESSING
TRIGGERS            CHECK CREDIT RATING
TRIGGERS            MAILING TO CUST
EXTRACTS            CUST ORDER
EXTRACTS            CUST FILE

ACTIVITY            NEW CUST PROCESSING
EXECUTED-BY         CREDIT CLERK
STIMULATED          INTERNALLY
PERFORMED-BY        CREDIT DEPT
EXTRACTS            NEW CUST MSG
GENERATES           CUST FILE
TRIGGERS            CHECK CREDIT RATING

ACTIVITY            CHECK CREDIT RATING
EXECUTED-BY         SALES CLERK 1
STIMULATED          INTERNALLY
PERFORMED-BY        SALES DEPT
EXTRACTS            CUST ORDER
EXTRACTS            CUST FILE
GENERATES           BAD CREDIT RATING MSG
TRIGGERS            CHECK CREDIT CEILING
TRIGGERS            MAILING TO CUST

REPORT              BAD CREDIT RATING MSG
GENERATED-BY        SALES CLERK 1
RECEIVED-BY         MAIL ROOM

ACTIVITY            CHECK CREDIT CEILING
EXECUTED-BY         SALES CLERK 1
STIMULATED          INTERNALLY
PROCESS             DOES ORDER EXCEED LIMIT?
EXTRACTS            CUST ORDER
EXTRACTS            CUST FILE
PERFORMED-BY        SALES DEPT
GENERATES           CREDIT CEILING EXCEEDED MSG
TRIGGERS            CHECK QUAN ON HAND
TRIGGERS            MAILING TO CUST

REPORT              CREDIT CEILING EXCEEDED MSG
RECEIVED-BY         MAIL ROOM

ACTIVITY            CHECK QUAN ON HAND
EXECUTED-BY         SALES CLERK 2
STIMULATED          INTERNALLY
PROCESS             IS ORDER > QOH?
PERFORMED-BY        SALES DEPT
EXTRACTS            CUST ORDER
EXTRACTS            FIN GOODS INVEN FILE
GENERATES           CUST BACKORDER FILE
TRIGGERS            PRODUCTION
TRIGGERS            CUST INVOICE PREP
TRIGGERS            SHIPPING NOTICE PREP
TRIGGERS            PACKING SLIP PREP
TRIGGERS            UPDATE SALES FILE

ACTIVITY            CUST INVOICE PREP
EXECUTED-BY         SALES CLERK 2
```

```
STIMULATED          INTERNALLY
PERFORMED-BY        SALES DEPT
GENERATES           CUST INVOICE
TRIGGERS            CUST INVOICE APPROVAL

ACTIVITY            CUST INVOICE APPROVAL
EXECUTED-BY         A/R CLERK
STIMULATED          INTERNALLY
PERFORMED-BY        ACCOUNTING DEPT
TRIGGERS            MAILING TO CUST
EXTRACTS            CUST INVOICE
GENERATES           CUST INVOICE

REPORT              CUST INVOICE
RECEIVED-BY         MAIL ROOM

ACTIVITY            SHIPPING NOTICE PREP
EXECUTED-BY         SALES CLERK 2
STIMULATED          INTERNALLY
PERFORMED-BY        SALES DEPT
GENERATES           SHIPPING NOTICE FILE
TRIGGERS            SHIPPING

ACTIVITY            PACKING SLIP PREP
EXECUTED-BY         SALES CLERK 2
STIMULATED          INTERNALLY
PERFORMED-BY        SALES DEPT
GENERATES           PACKING SLIP
TRIGGERS            PACK PRODUCT

ACTIVITY            UPDATE SALES FILE
EXECUTED-BY         SALES CLERK 2
STIMULATED          INTERNALLY
PERFORMED-BY        SALES DEPT
GENERATES           SALES FILE

        ***** CUST PAYMENT SLIP *****

REPORT              CUST PAYMENT SLIP
RECEIVED-BY         ACCOUNTING DEPT
PREPARED-BY         MAIL ROOM

ACTIVITY            CASH RECEIPTS
EXECUTED-BY         A/R CLERK
STIMULATED          EXTERNALLY
PERFORMED-BY        ACCOUNTING DEPT
EXTRACTS            CUST PAYMENT SLIP
EXTRACTS            CUST FILE
GENERATES           CUST FILE
GENERATES           UNLOCATED ACCOUNT MSG

REPORT              ACCOUNT BALANCE NEG MSG
RECEIVED-BY         MAIL ROOM

REPORT              UNLOCATED ACCOUNT MSG
RECEIVED-BY         MAIL ROOM

        ***** RECEIVING REPORT *****

REPORT              PURCHASE ORDER
PREPARED-BY         PURCHASING DEPT
```

```
RECEIVED-BY         RECEIVING DEPT

REPORT              RECEIVING REPORT
PREPARED-BY         RECEIVING DEPT

ACTIVITY            RECEIVING
EXECUTED-BY         RECEIVING CLERK 1
STIMULATED          INTERNALLY
PERFORMED-BY        RECEIVING DEPT
EXTRACTS            RECEIVING REPORT
EXTRACTS            VENDOR FILE
EXTRACTS            PURCHASE ORDER #2
GENERATES           PARTS RECEIPT DISCREPANCY REPORT
GENERATES           PARTS INVEN FILE
TRIGGERS            PAYMENT AUTH PREP
TRIGGERS            HANDLE PARTS DISCREPANCY

ACTIVITY            PAYMENT AUTH PREP
EXECUTED-BY         RECEIVING CLERK 2
STIMULATED          INTERNALLY
PERFORMED-BY        RECEIVING DEPT
GENERATES           PAYMENT AUTH
TRIGGERS            VENDOR PAYMENTS

REPORT              PAYMENT AUTH
RECEIVED-BY         ACCOUNTING DEPT

REPORT              PARTS RECEIPT DISCREPANCY REPORT
RECEIVED-BY         PURCHASING DEPT

ACTIVITY            VENDOR PAYMENTS
EXECUTED-BY         A/P CLERK
STIMULATED          INTERNALLY
PERFORMED-BY        ACCOUNTING DEPT
EXTRACTS            PAYMENT AUTH
GENERATES           VENDOR CHECK
TRIGGERS            MAILING TO VENDOR

REPORT              VENDOR CHECK
RECEIVED-BY         MAIL ROOM

        ***** PURCHASE INVOICE *****

REPORT              PURCHASE INVOICE
PREPARED-BY         VENDOR
RECEIVED-BY         ACCOUNTING DEPT
EXTRACTED-BY        PURCHASE INVOICE VERIFICATION

ACTIVITY            PURCHASE INVOICE VERIFICATION
EXECUTED-BY         A/P CLERK
STIMULATED          EXTERNALLY
PROCESS             SEE IF ALL INVOICED ITEMS WERE ORDERED
TRIGGERS            VENDOR PAYMENTS
PERFORMED-BY        ACCOUNTING DEPT
EXTRACTS            PURCHASE ORDER FILE
GENERATES           VENDOR FILE

        ***** PARTS REQUISITION *****

REPORT              PARTS REQUISITION
PREPARED-BY         PRODUCTION DEPT
```

While loading the data base, the loader routine verified whether object values were used consistently. This process caught the following types of errors, which were subsequently corrected:

EMPLOYEE was being used both as a DEPARTMENT and a POSITION. It was determined that EMPLOYEE was an ill-defined term, and it was removed.

Re: POSITION CREDIT CLERK / PERFORMS NEW CUST PROCESSING: PERFORMS is a relation between a DEPARTMENT and an ACTIVITY. The desired relation between this POSITION and its ACTIVITY is EXECUTED. There are a few pairs of relations that are easily confused, and these were the cause of most errors.

After the data base is loaded, an organized listing of the data base's contents is useful for evaluating its completeness and consistency (refer to Section 4.3.3). A common error not caught by the loader is the misspelling of object values. An alphabetical listing of object values makes it easy to visually detect these misspellings. Some inconsistent naming was found on previous iterations, were subsequently corrected, and do not appear in the following listing that indicates all attributes and relations of each object.

The following formatted listing indicates all attributes and relations of each object.

```
LOGIN 3305/10555

U of Ariz  6.03 KL  7:02 P.M. Wed Mar 2
Job 72  TTY243  User ARTIE
Balance = $147.11

.RUN FORMAT

REPORT FILE?
FORMAT.RPT
DATA BASE FILE?
Z
SORTED BY TYPE OR ALPHABETICALLY? (T OR A)
A
LISTING OF OBJECTS ONLY? (Y OR N)
N
STOP

END OF EXECUTION
CPU TIME: 32.45 ELAPSED TIME: 13:18.55
EXIT

.PRINT/DEL/FORMS:WN FORMAT.RPT
Total of 75 blocks in 1 file in LPT request

.K/F
Job 72, User [3305,10555]  Logged off TTY243   1917  2-Mar-77
Runtime 33.09 Sec; Session Charge $ 3.29
```

```
POSITION            A/P CLERK
EMPLOYED-BY         ACCOUNTING DEPT
EXECUTES            PURCHASE INVOICE VER
EXECUTES            VENDOR PAYMENTS

REPORT              A/P REPORT
INITIATED           INTERNALLY
GENERATED-BY        A/P REPORTING
RECEIVED-BY         ACCOUNTING DEPT

ACTIVITY            A/P REPORTING
PROCESS             COPY INFO FROM VENDO
STIMULATED          INTERNALLY
EXECUTED-BY         REPORT CLERK
EXTRACTS            VENDOR FILE
GENERATES           A/P REPORT
PERFORMED-BY        ACCOUNTING DEPT
TRIGGERED-BY        ACCOUNTING DEPT REQU

POSITION            A/R CLERK
EMPLOYED-BY         ACCOUNTING DEPT
EXECUTES            CASH RECEIPTS
EXECUTES            CUST INVOICE APPROVA

REPORT              A/R REPORT
INITIATED           INTERNALLY
GENERATED-BY        A/R REPORTING
RECEIVED-BY         ACCOUNTING DEPT

ACTIVITY            A/R REPORTING
PROCESS             COPY INFO FROM CUST
STIMULATED          INTERNALLY
EXECUTED-BY         REPORT CLERK
EXTRACTS            CUST FILE
GENERATES           A/R REPORT
PERFORMED-BY        ACCOUNTING DEPT
TRIGGERED-BY        ACCOUNTING DEPT REQU

REPORT              ACCOUNT BALANCE NEG
INITIATED           EXTERNALLY
EXTRACTED-BY        MAILING TO CUST
GENERATED-BY        CASH RECEIPTS
RECEIVED-BY         MAIL ROOM

POSITION            ACCOUNTANT 1
EMPLOYED-BY         ACCOUNTING DEPT
EXECUTES            ANNUAL INVEN

DEPARTMENT          ACCOUNTING DEPT
EMPLOYS             A/R CLERK
EMPLOYS             A/P CLERK
EMPLOYS             PAYROLL CLERK
EMPLOYS             ACCOUNTANT 1
EMPLOYS             REPORT CLERK
PERFORMS            PROFIT ANALYSIS GENE
PERFORMS            EMPL TAX REPORT GENE
PERFORMS            PRODUCTION REPORTING
PERFORMS            SALES REPORTING
PERFORMS            ANNUAL INVEN
PERFORMS            WAREHOUSE REPORTING
PERFORMS            A/P REPORTING
PERFORMS            A/R REPORTING
PERFORMS            ACCOUNTING DEPT REQU
PERFORMS            COMPANY TAX REPORT G
PERFORMS            PAYCHECK PREP
PERFORMS            PURCHASE INVOICE VER
PERFORMS            VENDOR PAYMENTS
PERFORMS            CASH RECEIPTS
PERFORMS            CUST INVOICE APPROVA
PREPARES            TIME CARD
RECEIVES            CUST FILE
RECEIVES            FIN GOODS INVEN FILE
RECEIVES            SALES FILE
RECEIVES            VENDOR FILE
RECEIVES            PARTS INVEN FILE
RECEIVES            PROFIT ANALYSIS REPO
RECEIVES            INVEN STATUS REPORT
RECEIVES            A/P REPORT
RECEIVES            A/R REPORT
RECEIVES            COMPANY TAX REPORT
RECEIVES            EMPLOYEE TAX REPORT
RECEIVES            EMPLOYEE FILE
RECEIVES            OPERATING INFO FILE
RECEIVES            ANNUAL INVEN REPORT
RECEIVES            PURCHASE ORDER FILE
RECEIVES            TIME CARD
RECEIVES            PURCHASE INVOICE
RECEIVES            PAYMENT AUTH
RECEIVES            CUST PAYMENT SLIP
RECEIVES            CUST INVOICE

ACTIVITY            ACCOUNTING DEPT REQU
PERIOD              ON DEMAND
PROCESS             REPORTS ON DEMAND
STIMULATED          INTERNALLY
PERFORMED-BY        ACCOUNTING DEPT
TRIGGERS            WAREHOUSE REPORTING
TRIGGERS            PAYCHECK PREP
TRIGGERS            PROFIT ANALYSIS GENE
TRIGGERS            EMPL TAX REPORT GENE
TRIGGERS            ANNUAL INVEN
TRIGGERS            A/P REPORTING
TRIGGERS            A/R REPORTING
TRIGGERS            COMPANY TAX REPORT G

ACTIVITY            ANNUAL INVEN
PERIOD              ANNUALLY
STIMULATED          INTERNALLY
EXECUTED-BY         ACCOUNTANT 1
GENERATES           ANNUAL INVEN REPORT
INCOMPATIBLE-WITH   PARTS PICKING
INCOMPATIBLE-WITH   PACK PRODUCT
PERFORMED-BY        ACCOUNTING DEPT
TRIGGERED-BY        ACCOUNTING DEPT REQU
TRIGGERS            WAREHOUSE REPORTING

REPORT              ANNUAL INVEN REPORT
EXTRACTED-BY        WAREHOUSE REPORTING
GENERATED-BY        ANNUAL INVEN
RECEIVED-BY         ACCOUNTING DEPT
```

```
REPORT              BAD CREDIT RATING MS
INITIATED           EXTERNALLY
EXTRACTED-BY        MAILING TO CUST
GENERATED-BY        CHECK CREDIT RATING
RECEIVED-BY         MAIL ROOM

ACTIVITY            CASH RECEIPTS
STIMULATED          INTERNALLY
EXECUTED-BY         A/R CLERK
EXTRACTS            CUST FILE
EXTRACTS            CUST PAYMENT SLIP
GENERATES           ACCOUNT BALANCE NEG
GENERATES           UNLOCATED ACCOUNT MS
GENERATES           CUST FILE
INCOMPATIBLE-WITH   MAILING FROM CUST
PERFORMED-BY        ACCOUNTING DEPT
TRIGGERED-BY        MAILING FROM CUST
TRIGGERS            MAILING TO CUST

ACTIVITY            CHECK CREDIT CEILING
PROCESS             DOES ORDER EXCEED LI
STIMULATED          INTERNALLY
EXECUTED-BY         SALES CLERK 1
EXTRACTS            CUST FILE
EXTRACTS            CUST ORDER
GENERATES           CREDIT CEILING EXCEE
INCOMPATIBLE-WITH   CUST ORDER PROCESSIN
PERFORMED-BY        SALES DEPT
TRIGGERED-BY        CHECK CREDIT RATING
TRIGGERS            MAILING TO CUST
TRIGGERS            CHECK QUAN ON HAND

ACTIVITY            CHECK CREDIT RATING
STIMULATED          INTERNALLY
EXECUTED-BY         SALES CLERK 1
EXTRACTS            CUST FILE
EXTRACTS            CUST ORDER
GENERATES           BAD CREDIT RATING MS
INCOMPATIBLE-WITH   CUST ORDER PROCESSIN
PERFORMED-BY        SALES DEPT
TRIGGERED-BY        DETERMINE IF NEW CUS
TRIGGERED-BY        NEW CUST PROCESSING
TRIGGERS            MAILING TO CUST
TRIGGERS            CHECK CREDIT CEILING

ACTIVITY            CHECK QUAN ON HAND
PROCESS             IS ORDER > QOH?
STIMULATED          INTERNALLY
EXECUTED-BY         SALES CLERK 2
EXTRACTS            FIN GOODS INVEN FILE
EXTRACTS            CUST ORDER
GENERATES           CUST BACKORDER MSG#2
GENERATES           CUST BACKORDER MSG#1
GENERATES           CUST BACKORDER FILE
PERFORMED-BY        SALES DEPT
TRIGGERED-BY        CHECK CREDIT CEILING
TRIGGERS            UPDATE SALES FILE
TRIGGERS            PACKING SLIP PREP
TRIGGERS            SHIPPING NOTICE PREP
TRIGGERS            CUST INVOICE PREP
TRIGGERS            PRODUCTION
```

```
ACTIVITY            CHECK REORDER POINT
STIMULATED          INTERNALLY
EXECUTED-BY         PRODUCTION CLERK
EXTRACTS            PARTS INVEN FILE
EXTRACTS            PARTS REQUISITION
GENERATES           PARTS BACKORDER
GENERATES           PARTS PICKING TICKET
PERFORMED-BY        PRODUCTION DEPT
TRIGGERED-BY        PARTS REQUISITIONING
TRIGGERS            PURCHASING
TRIGGERS            PARTS PICKING

REPORT              COMPANY TAX REPORT
INITIATED           INTERNALLY
GENERATED-BY        COMPANY TAX REPORT G
RECEIVED-BY         ACCOUNTING DEPT

ACTIVITY            COMPANY TAX REPORT G
PROCESS             COPY TAXES ON OP INC
STIMULATED          INTERNALLY
EXECUTED-BY         REPORT CLERK
EXTRACTS            OPERATING INFO FILE
GENERATES           COMPANY TAX REPORT
PERFORMED-BY        ACCOUNTING DEPT
TRIGGERED-BY        ACCOUNTING DEPT REQU

REPORT              CREDIT CEILING EXCEE
INITIATED           EXTERNALLY
EXTRACTED-BY        MAILING TO CUST
GENERATED-BY        CHECK CREDIT CEILING
RECEIVED-BY         MAIL ROOM

POSITION            CREDIT CLERK
EMPLOYED-BY         CREDIT DEPT
EXECUTES            UPDATE CUST INFO
EXECUTES            NEW CUST PROCESSING

DEPARTMENT          CREDIT DEPT
EMPLOYS             CREDIT CLERK
PERFORMS            UPDATE CUST INFO
PERFORMS            NEW CUST PROCESSING
PREPARES            TIME CARD
RECEIVES            NEW CUST MSG
RECEIVES            CUST FILE
RECEIVES            CUST FILE CHANGES

DEPARTMENT          CUST
PREPARES            MAIL FROM CUST
RECEIVES            MAIL TO CUST

REPORT              CUST BACKORDER FILE
EXTRACTED-BY        CUST BACKORDER GENER
GENERATED-BY        CHECK QUAN ON HAND
RECEIVED-BY         PURCHASING DEPT

ACTIVITY            CUST BACKORDER GENER
PROCESS             CREATE BACKORDER REP
STIMULATED          INTERNALLY
EXECUTED-BY         PURCHASING CLERK
EXTRACTS            CUST BACKORDER FILE
```

```
GENERATES              CUST BACKORDER REPOR
PERFORMED-BY           PURCHASING DEPT
TRIGGERED-BY           PURCHASING DEPT REQU

REPORT                 CUST BACKORDER MSG#1
EXTRACTED-BY           PRODUCTION
GENERATED-BY           CHECK QUAN ON HAND
RECEIVED-BY            PRODUCTION DEPT

REPORT                 CUST BACKORDER MSG#2
INITIATED              EXTERNALLY
EXTRACTED-BY           MAILING TO CUST
GENERATED-BY           CHECK QUAN ON HAND
RECEIVED-BY            MAIL ROOM

REPORT                 CUST BACKORDER REPOR
INITIATED              INTERNALLY
GENERATED-BY           CUST BACKORDER GENER
RECEIVED-BY            SALES DEPT

REPORT                 CUST FILE
EXTRACTED-BY           DETERMINE IF NEW CUS
EXTRACTED-BY           UPDATE CUST INFO
EXTRACTED-BY           CUST REPORTING
EXTRACTED-BY           A/R REPORTING
EXTRACTED-BY           CASH RECEIPTS
EXTRACTED-BY           CHECK CREDIT CEILING
EXTRACTED-BY           CHECK CREDIT RATING
GENERATED-BY           NEW CUST PROCESSING
GENERATED-BY           UPDATE CUST INFO
GENERATED-BY           CASH RECEIPTS
RECEIVED-BY            SALES DEPT
RECEIVED-BY            ACCOUNTING DEPT
RECEIVED-BY            CREDIT DEPT

REPORT                 CUST FILE CHANGES
EXTRACTED-BY           UPDATE CUST INFO
GENERATED-BY           DETERMINE CUST CHANG
PREPARED-BY            SALES DEPT
RECEIVED-BY            CREDIT DEPT

REPORT                 CUST INVOICE
INITIATED              EXTERNALLY
EXTRACTED-BY           MAILING TO CUST
EXTRACTED-BY           CUST INVOICE APPROVA
GENERATED-BY           CUST INVOICE PREP
GENERATED-BY           CUST INVOICE APPROVA
RECEIVED-BY            ACCOUNTING DEPT
RECEIVED-BY            MAIL ROOM

ACTIVITY               CUST INVOICE APPROVA
STIMULATED             INTERNALLY
EXECUTED-BY            A/R CLERK
EXTRACTS               CUST INVOICE
GENERATES              CUST INVOICE
INCOMPATIBLE-WITH      MAILING TO CUST
INCOMPATIBLE-WITH      CUST INVOICE PREP
INCOMPATIBLE-WITH      SHIPPING
INCOMPATIBLE-WITH      PACK PRODUCT
INCOMPATIBLE-WITH      CUST ORDER PROCESSIN
PERFORMED-BY           ACCOUNTING DEPT
```

```
TRIGGERED-BY         CUST INVOICE PREP
TRIGGERS             MAILING TO CUST

ACTIVITY             CUST INVOICE PREP
STIMULATED           INTERNALLY
EXECUTED-BY          SALES CLERK 2
EXTRACTS             CUST ORDER
GENERATES            CUST INVOICE
INCOMPATIBLE-WITH    CUST INVOICE APPROVA
INCOMPATIBLE-WITH    MAILING TO CUST
PERFORMED-BY         SALES DEPT
TRIGGERED-BY         CHECK QUAN ON HAND
TRIGGERS             CUST INVOICE APPROVA

REPORT               CUST ORDER
INITIATED            EXTERNALLY
EXTRACTED-BY         CUST ORDER PROCESSIN
EXTRACTED-BY         CUST INVOICE PREP
EXTRACTED-BY         SHIPPING NOTICE PREP
EXTRACTED-BY         PACKING SLIP PREP
EXTRACTED-BY         UPDATE SALES FILE
EXTRACTED-BY         CHECK QUAN ON HAND
EXTRACTED-BY         CHECK CREDIT CEILING
EXTRACTED-BY         CHECK CREDIT RATING
EXTRACTED-BY         DETERMINE IF NEW CUS
GENERATED-BY         CUST ORDER TAKING
GENERATED-BY         CUST ORDER PROCESSIN
RECEIVED-BY          SALES DEPT

ACTIVITY             CUST ORDER PROCESSIN
PROCESS              VERIFY ACCURACY & CO
STIMULATED           INTERNALLY
EXECUTED-BY          SALES CLERK 1
EXTRACTS             CUST ORDER
GENERATES            CUST ORDER
INCOMPATIBLE-WITH    CUST INVOICE APPROVA
INCOMPATIBLE-WITH    SHIPPING
INCOMPATIBLE-WITH    PACK PRODUCT
INCOMPATIBLE-WITH    CHECK CREDIT CEILING
INCOMPATIBLE-WITH    CHECK CREDIT RATING
PERFORMED-BY         SALES DEPT
TRIGGERED-BY         CUST ORDER TAKING
TRIGGERS             DETERMINE IF NEW CUS

ACTIVITY             CUST ORDER TAKING
STIMULATED           EXTERNALLY
EXECUTED-BY          SALESPERSON
GENERATES            CUST ORDER
PERFORMED-BY         SALES DEPT
TRIGGERS             CUST ORDER PROCESSIN

REPORT               CUST PAYMENT SLIP
INITIATED            EXTERNALLY
EXTRACTED-BY         CASH RECEIPTS
GENERATED-BY         MAILING FROM CUST
PREPARED-BY          MAIL ROOM
RECEIVED-BY          ACCOUNTING DEPT

REPORT               CUST REPORT
INITIATED            INTERNALLY
GENERATED-BY         CUST REPORTING
```

```
RECEIVED-BY          SALES DEPT

ACTIVITY             CUST REPORTING
STIMULATED           INTERNALLY
EXECUTED-BY          SALES CLERK 3
EXTRACTS             CUST FILE
GENERATES            CUST REPORT
PERFORMED-BY         SALES DEPT
TRIGGERED-BY         SALES DEPT REQUEST

ACTIVITY             DETERMINE CUST CHANG
STIMULATED           INTERNALLY
EXECUTED-BY          SALES CLERK 1
GENERATES            CUST FILE CHANGES
PERFORMED-BY         SALES DEPT
TRIGGERED-BY         SALES DEPT REQUEST
TRIGGERS             UPDATE CUST INFO

ACTIVITY             DETERMINE IF NEW CUS
STIMULATED           INTERNALLY
EXECUTED-BY          SALES CLERK 1
EXTRACTS             CUST FILE
EXTRACTS             CUST ORDER
GENERATES            NEW CUST MSG
PERFORMED-BY         SALES DEPT
TRIGGERED-BY         CUST ORDER PROCESSIN
TRIGGERS             MAILING TO CUST
TRIGGERS             CHECK CREDIT RATING
TRIGGERS             NEW CUST PROCESSING

ACTIVITY             EMPL TAX REPORT GENE
PROCESS              COPY TAX INFO TO PRI
STIMULATED           INTERNALLY
EXECUTED-BY          REPORT CLERK
EXTRACTS             EMPLOYEE FILE
GENERATES            EMPLOYEE TAX REPORT
PERFORMED-BY         ACCOUNTING DEPT
TRIGGERED-BY         ACCOUNTING DEPT REQU

REPORT               EMPLOYEE CHANGE FORM
EXTRACTED-BY         EMPLOYEE FILE MAINTE
RECEIVED-BY          PERSONNEL DEPT

REPORT               EMPLOYEE FILE
EXTRACTED-BY         PAYCHECK PREP
EXTRACTED-BY         EMPL TAX REPORT GENE
EXTRACTED-BY         EMPLOYEE FILE MAINTE
GENERATED-BY         PAYCHECK PREP
GENERATED-BY         EMPLOYEE FILE MAINTE
RECEIVED-BY          PERSONNEL DEPT
RECEIVED-BY          ACCOUNTING DEPT

ACTIVITY             EMPLOYEE FILE MAINTE
STIMULATED           INTERNALLY
EXECUTED-BY          PERSONNEL CLERK
EXTRACTS             EMPLOYEE CHANGE FORM
EXTRACTS             EMPLOYEE FILE
GENERATES            EMPLOYEE FILE
INCOMPATIBLE-WITH    PAYCHECK PREP
PERFORMED-BY         PERSONNEL DEPT
TRIGGERED-BY         PERSONNEL DEPT REQUE
```

```
REPORT              EMPLOYEE TAX REPORT
INITIATED           INTERNALLY
GENERATED-BY        EMPL TAX REPORT GENE
RECEIVED-BY         ACCOUNTING DEPT

REPORT              FIN GOODS INVEN FILE
EXTRACTED-BY        CHECK QUAN ON HAND
EXTRACTED-BY        PRODUCTION REPORTING
EXTRACTED-BY        PACK PRODUCT
EXTRACTED-BY        WAREHOUSE REPORTING
GENERATED-BY        PRODUCTION
GENERATED-BY        PACK PRODUCT
RECEIVED-BY         ACCOUNTING DEPT
RECEIVED-BY         SALES DEPT
RECEIVED-BY         WAREHOUSE

ACTIVITY            HANDLE PARTS DISCREP
STIMULATED          INTERNALLY
EXTRACTS            PARTS RECEIPT DISCRE
PERFORMED-BY        PURCHASING DEPT
TRIGGERED-BY        RECEIVING

REPORT              INVEN STATUS REPORT
INITIATED           INTERNALLY
GENERATED-BY        WAREHOUSE REPORTING
RECEIVED-BY         ACCOUNTING DEPT

POSITION            MACHINE OPERATOR
EMPLOYED-BY         PRODUCTION DEPT
EXECUTES            PARTS REQUISITIONING
EXECUTES            PRODUCTION

POSITION            MAIL CLERK
EMPLOYED-BY         MAIL ROOM
EXECUTES            MAILING FROM VENDOR
EXECUTES            MAILING FROM CUST
EXECUTES            MAILING TO CUST
EXECUTES            MAILING TO VENDOR

REPORT              MAIL FROM CUST
EXTRACTED-BY        MAILING FROM CUST
PREPARED-BY         CUST
RECEIVED-BY         MAIL ROOM

REPORT              MAIL FROM VENDOR
EXTRACTED-BY        MAILING FROM VENDOR
PREPARED-BY         VENDOR
RECEIVED-BY         MAIL ROOM

DEPARTMENT          MAIL ROOM
EMPLOYS             MAIL CLERK
PERFORMS            MAILING FROM VENDOR
PERFORMS            MAILING FROM CUST
PERFORMS            MAILING TO CUST
PERFORMS            MAILING TO VENDOR
PREPARES            TIME CARD
PREPARES            CUST PAYMENT SLIP
RECEIVES            MAIL FROM CUST
RECEIVES            MAIL FROM VENDOR
RECEIVES            CUST BACKORDER MSG#2
RECEIVES              PURCHASE ORDER #1
RECEIVES              SHIPPED GOODS
RECEIVES              VENDOR CHECK
RECEIVES              UNLOCATED ACCOUNT MS
RECEIVES              ACCOUNT BALANCE NEG
RECEIVES              CUST INVOICE
RECEIVES              CREDIT CEILING EXCEE
RECEIVES              BAD CREDIT RATING MS

REPORT                MAIL TO CUST
GENERATED-BY          MAILING TO CUST
RECEIVED-BY           CUST

REPORT                MAIL TO VENDOR
GENERATED-BY          MAILING TO VENDOR
RECEIVED-BY           VENDOR

ACTIVITY              MAILING FROM CUST
STIMULATED            EXTERNALLY
EXECUTED-BY           MAIL CLERK
EXTRACTS              MAIL FROM CUST
GENERATES             CUST PAYMENT SLIP
INCOMPATIBLE-WITH     CASH RECEIPTS
INCOMPATIBLE-WITH     MAILING TO CUST
PERFORMED-BY          MAIL ROOM
TRIGGERS              CASH RECEIPTS

ACTIVITY              MAILING FROM VENDOR
STIMULATED            EXTERNALLY
EXECUTED-BY           MAIL CLERK
EXTRACTS              MAIL FROM VENDOR
GENERATES             PURCHASE INVOICE
INCOMPATIBLE-WITH     PARTS REQUISITIONING
INCOMPATIBLE-WITH     PURCHASE INVOICE VER
INCOMPATIBLE-WITH     VENDOR PAYMENTS
INCOMPATIBLE-WITH     PAYMENT AUTH PREP
INCOMPATIBLE-WITH     MAILING TO VENDOR
PERFORMED-BY          MAIL ROOM
TRIGGERS              PURCHASE INVOICE VER

ACTIVITY              MAILING TO CUST
STIMULATED            INTERNALLY
EXECUTED-BY           MAIL CLERK
EXTRACTS              ACCOUNT BALANCE NEG
EXTRACTS              CUST BACKORDER MSG#2
EXTRACTS              UNLOCATED ACCOUNT MS
EXTRACTS              CUST INVOICE
EXTRACTS              CREDIT CEILING EXCEE
EXTRACTS              BAD CREDIT RATING MS
GENERATES             MAIL TO CUST
INCOMPATIBLE-WITH     MAILING FROM CUST
INCOMPATIBLE-WITH     CUST INVOICE APPROVA
INCOMPATIBLE-WITH     CUST INVOICE PREP
PERFORMED-BY          MAIL ROOM
TRIGGERED-BY          DETERMINE IF NEW CUS
TRIGGERED-BY          CASH RECEIPTS
TRIGGERED-BY          CUST INVOICE APPROVA
TRIGGERED-BY          CHECK CREDIT CEILING
TRIGGERED-BY          CHECK CREDIT RATING

ACTIVITY              MAILING TO VENDOR
```

```
STIMULATED             INTERNALLY
EXECUTED-BY            MAIL CLERK
EXTRACTS               PURCHASE ORDER #1
EXTRACTS               VENDOR CHECK
GENERATES              MAIL TO VENDOR.
INCOMPATIBLE-WITH      PARTS REQUISITIONING
INCOMPATIBLE-WITH      PURCHASE INVOICE VER
INCOMPATIBLE-WITH      VENDOR PAYMENTS
INCOMPATIBLE-WITH      PAYMENT AUTH PREP
INCOMPATIBLE-WITH      MAILING FROM VENDOR
PERFORMED-BY           MAIL ROOM
TRIGGERED-BY           VENDOR PAYMENTS
TRIGGERED-BY           PURCHASING

REPORT                 NEW CUST MSG
INITIATED              EXTERNALLY
EXTRACTED-BY           NEW CUST PROCESSING
GENERATED-BY           DETERMINE IF NEW CUS
RECEIVED-BY            CREDIT DEPT

ACTIVITY               NEW CUST PROCESSING
STIMULATED             INTERNALLY
EXECUTED-BY            CREDIT CLERK
EXTRACTS               NEW CUST MSG
GENERATES              CUST FILE
PERFORMED-BY           CREDIT DEPT
TRIGGERED-BY           DETERMINE IF NEW CUS
TRIGGERS               CHECK CREDIT RATING

REPORT                 OPERATING INFO FILE
EXTRACTED-BY           COMPANY TAX REPORT G
EXTRACTED-BY           PROFIT ANALYSIS GENE
RECEIVED-BY            ACCOUNTING DEPT

ACTIVITY               PACK PRODUCT
STIMULATED             INTERNALLY
EXECUTED-BY            WAREHOUSE CLERK
EXTRACTS               FIN GOODS INVEN FILE
EXTRACTS               PACKING SLIP
GENERATES              PACKED PRODUCT
GENERATES              FIN GOODS INVEN FILE
INCOMPATIBLE-WITH      PARTS PICKING
INCOMPATIBLE-WITH      ANNUAL INVEN
INCOMPATIBLE-WITH      CUST INVOICE APPROVA
INCOMPATIBLE-WITH      SHIPPING
INCOMPATIBLE-WITH      CUST ORDER PROCESSIN
PERFORMED-BY           WAREHOUSE
TRIGGERED-BY           PACKING SLIP PREP
TRIGGERS               SHIPPING

REPORT                 PACKED PRODUCT
EXTRACTED-BY           SHIPPING
GENERATED-BY           PACK PRODUCT
RECEIVED-BY            SHIPPING DEPT

REPORT                 PACKING SLIP
INITIATED              EXTERNALLY
EXTRACTED-BY           PACK PRODUCT
GENERATED-BY           PACKING SLIP PREP
RECEIVED-BY            WAREHOUSE
```

```
ACTIVITY                PACKING SLIP PREP
STIMULATED              INTERNALLY
EXECUTED-BY             SALES CLERK 2
EXTRACTS                CUST ORDER
GENERATES               PACKING SLIP
PERFORMED-BY            SALES DEPT
TRIGGERED-BY            CHECK QUAN ON HAND
TRIGGERS                PACK PRODUCT

REPORT                  PARTS BACKORDER
INITIATED               EXTERNALLY
EXTRACTED-BY            PURCHASING
GENERATED-BY            CHECK REORDER POINT
PREPARED-BY             PRODUCTION DEPT
RECEIVED-BY             PURCHASING DEPT

REPORT                  PARTS INVEN FILE
EXTRACTED-BY            CHECK REORDER POINT
EXTRACTED-BY            PARTS REQUISITIONING
EXTRACTED-BY            WAREHOUSE REPORTING
EXTRACTED-BY            PARTS PICKING
GENERATED-BY            RECEIVING
GENERATED-BY            PARTS REQUISITIONING
GENERATED-BY            PARTS PICKING
RECEIVED-BY             ACCOUNTING DEPT
RECEIVED-BY             WAREHOUSE
RECEIVED-BY             PRODUCTION DEPT

ACTIVITY                PARTS PICKING
STIMULATED              INTERNALLY
EXECUTED-BY             WAREHOUSE CLERK
EXTRACTS                PARTS INVEN FILE
EXTRACTS                PARTS PICKING TICKET
GENERATES               PICKED PARTS
GENERATES               PARTS INVEN FILE
INCOMPATIBLE-WITH       PACK PRODUCT
INCOMPATIBLE-WITH       PARTS REQUISITIONING
INCOMPATIBLE-WITH       ANNUAL INVEN
PERFORMED-BY            WAREHOUSE
TRIGGERED-BY            CHECK REORDER POINT
TRIGGERS                PRODUCTION

REPORT                  PARTS PICKING TICKET
INITIATED               EXTERNALLY
EXTRACTED-BY            PARTS PICKING
GENERATED-BY            CHECK REORDER POINT
RECEIVED-BY             WAREHOUSE

REPORT                  PARTS RECEIPT DISCRE
INITIATED               EXTERNALLY
EXTRACTED-BY            HANDLE PARTS DISCREP
GENERATED-BY            RECEIVING
RECEIVED-BY             PURCHASING DEPT

REPORT                  PARTS REQUISITION
INITIATED               EXTERNALLY
EXTRACTED-BY            CHECK REORDER POINT
GENERATED-BY            PARTS REQUISITIONING
PREPARED-BY             PRODUCTION DEPT
RECEIVED-BY             PRODUCTION DEPT
```

```
ACTIVITY             PARTS REQUISITIONING
STIMULATED           INTERNALLY
EXECUTED-BY          MACHINE OPERATOR
EXTRACTS             PARTS INVEN FILE
GENERATES            PARTS INVEN FILE
GENERATES            PARTS REQUISITION
INCOMPATIBLE-WITH    PARTS PICKING
INCOMPATIBLE-WITH    MAILING TO VENDOR
INCOMPATIBLE-WITH    PURCHASE INVOICE VER
INCOMPATIBLE-WITH    VENDOR PAYMENTS
INCOMPATIBLE-WITH    PAYMENT AUTH PREP
INCOMPATIBLE-WITH    MAILING FROM VENDOR
PERFORMED-BY         PRODUCTION DEPT
TRIGGERED-BY         PRODUCTION
TRIGGERS             CHECK REORDER POINT

REPORT               PAYCHECK
INITIATED            INTERNALLY
GENERATED-BY         PAYCHECK PREP

ACTIVITY             PAYCHECK PREP
STIMULATED           INTERNALLY
EXECUTED-BY          PAYROLL CLERK
EXTRACTS             TIME CARD
EXTRACTS             EMPLOYEE FILE
GENERATES            RECORD OF EARNINGS
GENERATES            PAYCHECK
GENERATES            EMPLOYEE FILE
INCOMPATIBLE-WITH    EMPLOYEE FILE MAINTE
PERFORMED-BY         ACCOUNTING DEPT
TRIGGERED-BY         ACCOUNTING DEPT REQU

REPORT               PAYMENT AUTH
INITIATED            EXTERNALLY
EXTRACTED-BY         VENDOR PAYMENTS
GENERATED-BY         PAYMENT AUTH PREP
RECEIVED-BY          ACCOUNTING DEPT

ACTIVITY             PAYMENT AUTH PREP
STIMULATED           INTERNALLY
EXECUTED-BY          RECEIVING CLERK 2
EXTRACTS             PURCHASE ORDER #2
EXTRACTS             RECEIVING REPORT
GENERATES            PAYMENT AUTH
INCOMPATIBLE-WITH    PARTS REQUISITIONING
INCOMPATIBLE-WITH    PURCHASE INVOICE VER
INCOMPATIBLE-WITH    VENDOR PAYMENTS
INCOMPATIBLE-WITH    MAILING TO VENDOR
INCOMPATIBLE-WITH    MAILING FROM VENDOR
PERFORMED-BY         RECEIVING DEPT
TRIGGERED-BY         RECEIVING
TRIGGERS             VENDOR PAYMENTS

POSITION             PAYROLL CLERK
EMPLOYED-BY          ACCOUNTING DEPT
EXECUTES             PAYCHECK PREP

POSITION             PERSONNEL CLERK
EMPLOYED-BY          PERSONNEL DEPT
EXECUTES             EMPLOYEE FILE MAINTE
```

```
DEPARTMENT          PERSONNEL DEPT
EMPLOYS             PERSONNEL CLERK
PERFORMS            PERSONNEL DEPT REQUE
PERFORMS            EMPLOYEE FILE MAINTE
PREPARES            TIME CARD
RECEIVES            EMPLOYEE FILE
RECEIVES            EMPLOYEE CHANGE FORM

ACTIVITY            PERSONNEL DEPT REQUE
STIMULATED          INTERNALLY
PERFORMED-BY        PERSONNEL DEPT
TRIGGERS            EMPLOYEE FILE MAINTE

REPORT              PICKED PARTS
EXTRACTED-BY        PRODUCTION
GENERATED-BY        PARTS PICKING
RECEIVED-BY         PRODUCTION DEPT

ACTIVITY            PRODUCTION
STIMULATED          INTERNALLY
EXECUTED-BY         MACHINE OPERATOR
EXTRACTS            CUST BACKORDER MSG#1
EXTRACTS            PICKED PARTS
GENERATES           FIN GOODS INVEN FILE
PERFORMED-BY        PRODUCTION DEPT
TRIGGERED-BY        CHECK QUAN ON HAND
TRIGGERED-BY        PARTS PICKING
TRIGGERS            PARTS REQUISITIONING

POSITION            PRODUCTION CLERK
EMPLOYED-BY         PRODUCTION DEPT
EXECUTES            CHECK REORDER POINT

DEPARTMENT          PRODUCTION DEPT
EMPLOYS             PRODUCTION CLERK
EMPLOYS             MACHINE OPERATOR
PERFORMS            PRODUCTION DEPT REQU
PERFORMS            PRODUCTION
PERFORMS            CHECK REORDER POINT
PERFORMS            PARTS REQUISITIONING
PREPARES            TIME CARD
PREPARES            PARTS BACKORDER
PREPARES            PARTS REQUISITION
RECEIVES            PICKED PARTS
RECEIVES            PARTS REQUISITION
RECEIVES            PARTS INVEN FILE
RECEIVES            CUST BACKORDER MSG#1
RECEIVES            PRODUCTION REPORT

ACTIVITY            PRODUCTION DEPT REQU
STIMULATED          INTERNALLY
PERFORMED-BY        PRODUCTION DEPT
TRIGGERS            PRODUCTION REPORTING

REPORT              PRODUCTION REPORT
INITIATED           INTERNALLY
GENERATED-BY        PRODUCTION REPORTING
RECEIVED-BY         PRODUCTION DEPT

ACTIVITY            PRODUCTION REPORTING
STIMULATED          INTERNALLY
```

```
EXECUTED-BY          REPORT CLERK
EXTRACTS             FIN GOODS INVEN FILE
GENERATES            PRODUCTION REPORT
PERFORMED-BY         ACCOUNTING DEPT
TRIGGERED-BY         PRODUCTION DEPT REQU

ACTIVITY             PROFIT ANALYSIS GENE
PERIOD               MONTHLY
STIMULATED           INTERNALLY
EXECUTED-BY          REPORT CLERK
EXTRACTS             OPERATING INFO FILE
GENERATES            PROFIT ANALYSIS REPO
PERFORMED-BY         ACCOUNTING DEPT
TRIGGERED-BY         ACCOUNTING DEPT REQU

REPORT               PROFIT ANALYSIS REPO
INITIATED            INTERNALLY
GENERATED-BY         PROFIT ANALYSIS GENE
RECEIVED-BY          ACCOUNTING DEPT

REPORT               PURCHASE INVOICE
INITIATED            EXTERNALLY
EXTRACTED-BY         PURCHASE INVOICE VER
GENERATED-BY         MAILING FROM VENDOR
PREPARED-BY          VENDOR
RECEIVED-BY          ACCOUNTING DEPT

ACTIVITY             PURCHASE INVOICE VER
PROCESS              SEE IF ALL INVOICED
STIMULATED           INTERNALLY
EXECUTED-BY          A/P CLERK
EXTRACTS             PURCHASE ORDER FILE
EXTRACTS             PURCHASE INVOICE
GENERATES            VENDOR FILE
INCOMPATIBLE-WITH    PARTS REQUISITIONING
INCOMPATIBLE-WITH    MAILING TO VENDOR
INCOMPATIBLE-WITH    VENDOR PAYMENTS
INCOMPATIBLE-WITH    PAYMENT AUTH PREP
INCOMPATIBLE-WITH    MAILING FROM VENDOR
PERFORMED-BY         ACCOUNTING DEPT
TRIGGERED-BY         MAILING FROM VENDOR
TRIGGERS             VENDOR PAYMENTS

REPORT               PURCHASE ORDER #1
EXTRACTED-BY         MAILING TO VENDOR
GENERATED-BY         PURCHASING
RECEIVED-BY          MAIL ROOM

REPORT               PURCHASE ORDER #2
EXTRACTED-BY         RECEIVING
EXTRACTED-BY         PAYMENT AUTH PREP
GENERATED-BY         PURCHASING
RECEIVED-BY          RECEIVING DEPT

REPORT               PURCHASE ORDER FILE
EXTRACTED-BY         PURCHASE INVOICE VER
EXTRACTED-BY         WAREHOUSE REPORTING
GENERATED-BY         PURCHASING
RECEIVED-BY          ACCOUNTING DEPT

ACTIVITY             PURCHASING
```

```
STIMULATED          INTERNALLY
EXECUTED-BY         PURCHASING CLERK
EXTRACTS            VENDOR FILE
EXTRACTS            PARTS BACKORDER
GENERATES           PURCHASE ORDER #2
GENERATES           PURCHASE ORDER #1
GENERATES           PURCHASE ORDER FILE
INCOMPATIBLE-WITH   SHIPPING
INCOMPATIBLE-WITH   RECEIVING
PERFORMED-BY        PURCHASING DEPT
TRIGGERED-BY        CHECK REORDER POINT
TRIGGERS            RECEIVING
TRIGGERS            MAILING TO VENDOR

POSITION            PURCHASING CLERK
EMPLOYED-BY         PURCHASING DEPT
EXECUTES            CUST BACKORDER GENER
EXECUTES            PURCHASING

DEPARTMENT          PURCHASING DEPT
EMPLOYS             PURCHASING CLERK
PERFORMS            PURCHASING DEPT REQU
PERFORMS            CUST BACKORDER GENER
PERFORMS            PURCHASING
PERFORMS            HANDLE PARTS DISCREP
PREPARES            TIME CARD
RECEIVES            CUST BACKORDER FILE
RECEIVES            VENDOR FILE
RECEIVES            PARTS BACKORDER
RECEIVES            PARTS RECEIPT DISCRE

ACTIVITY            PURCHASING DEPT REQU
STIMULATED          INTERNALLY
PERFORMED-BY        PURCHASING DEPT
TRIGGERS            CUST BACKORDER GENER

ACTIVITY            RECEIVING
STIMULATED          INTERNALLY
EXECUTED-BY         RECEIVING CLERK 1
EXTRACTS            PURCHASE ORDER #2
EXTRACTS            VENDOR FILE
EXTRACTS            RECEIVING REPORT
GENERATES           PARTS INVEN FILE
GENERATES           PARTS RECEIPT DISCRE
INCOMPATIBLE-WITH   PURCHASING
INCOMPATIBLE-WITH   SHIPPING
PERFORMED-BY        RECEIVING DEPT
TRIGGERED-BY        PURCHASING
TRIGGERS            HANDLE PARTS DISCREP
TRIGGERS            PAYMENT AUTH PREP

POSITION            RECEIVING CLERK 1
EMPLOYED-BY         RECEIVING DEPT
EXECUTES            RECEIVING

POSITION            RECEIVING CLERK 2
EMPLOYED-BY         RECEIVING DEPT
EXECUTES            PAYMENT AUTH PREP

DEPARTMENT          RECEIVING DEPT
EMPLOYS             RECEIVING CLERK 2
EMPLOYS             RECEIVING CLERK 1
PERFORMS            PAYMENT AUTH PREP
PERFORMS            RECEIVING
PREPARES            TIME CARD
PREPARES            RECEIVING REPORT
RECEIVES            RECEIVING REPORT
RECEIVES            VENDOR FILE
RECEIVES            PURCHASE ORDER #2

REPORT              RECEIVING REPORT
INITIATED           EXTERNALLY
EXTRACTED-BY        RECEIVING
EXTRACTED-BY        PAYMENT AUTH PREP
PREPARED-BY         RECEIVING DEPT
RECEIVED-BY         RECEIVING DEPT

REPORT              RECORD OF EARNINGS
GENERATED-BY        PAYCHECK PREP

POSITION            REPORT CLERK
EMPLOYED-BY         ACCOUNTING DEPT
EXECUTES            WAREHOUSE REPORTING
EXECUTES            PROFIT ANALYSIS GENE
EXECUTES            EMPL TAX REPORT GENE
EXECUTES            COMPANY TAX REPORT G
EXECUTES            A/P REPORTING
EXECUTES            PRODUCTION REPORTING
EXECUTES            SALES REPORTING
EXECUTES            A/R REPORTING

POSITION            SALES CLERK 1
EMPLOYED-BY         SALES DEPT
EXECUTES            DETERMINE CUST CHANG
EXECUTES            CHECK CREDIT CEILING
EXECUTES            CHECK CREDIT RATING
EXECUTES            DETERMINE IF NEW CUS
EXECUTES            CUST ORDER PROCESSIN

POSITION            SALES CLERK 2
EMPLOYED-BY         SALES DEPT
EXECUTES            UPDATE SALES FILE
EXECUTES            PACKING SLIP PREP
EXECUTES            SHIPPING NOTICE PREP
EXECUTES            CUST INVOICE PREP
EXECUTES            CHECK QUAN ON HAND

POSITION            SALES CLERK 3
EMPLOYED-BY         SALES DEPT
EXECUTES            CUST REPORTING

DEPARTMENT          SALES DEPT
EMPLOYS             SALES CLERK 3
EMPLOYS             SALES CLERK 2
EMPLOYS             SALES CLERK 1
EMPLOYS             SALESPERSON
PERFORMS            SALES DEPT REQUEST
PERFORMS            DETERMINE CUST CHANG
PERFORMS            CUST REPORTING
PERFORMS            UPDATE SALES FILE
PERFORMS            PACKING SLIP PREP
PERFORMS            SHIPPING NOTICE PREP
PERFORMS            CUST INVOICE PREP
PERFORMS            CHECK QUAN ON HAND
PERFORMS            CHECK CREDIT CEILING
PERFORMS            CHECK CREDIT RATING
PERFORMS            DETERMINE IF NEW CUS
PERFORMS            CUST ORDER PROCESSIN
PERFORMS            CUST ORDER TAKING
PREPARES            TIME CARD
PREPARES            CUST FILE CHANGES
RECEIVES            CUST FILE
RECEIVES            FIN GOODS INVEN FILE
RECEIVES            CUST BACKORDER REPOR
RECEIVES            SALES REPORT
RECEIVES            CUST REPORT
RECEIVES            CUST ORDER

ACTIVITY            SALES DEPT REQUEST
STIMULATED          INTERNALLY
PERFORMED-BY        SALES DEPT
TRIGGERS            SALES REPORTING
TRIGGERS            DETERMINE CUST CHANG
TRIGGERS            CUST REPORTING

REPORT              SALES FILE
EXTRACTED-BY        SALES REPORTING
GENERATED-BY        UPDATE SALES FILE
RECEIVED-BY         ACCOUNTING DEPT

REPORT              SALES REPORT
INITIATED           INTERNALLY
GENERATED-BY        SALES REPORTING
RECEIVED-BY         SALES DEPT

ACTIVITY            SALES REPORTING
PERIOD              WEEKLY
STIMULATED          INTERNALLY
EXECUTED-BY         REPORT CLERK
EXTRACTS            SALES FILE
GENERATES           SALES REPORT
PERFORMED-BY        ACCOUNTING DEPT
TRIGGERED-BY        SALES DEPT REQUEST

POSITION            SALESPERSON
PERSON              1 OF 100 SALESPEOPLE
EMPLOYED-BY         SALES DEPT
EXECUTES            CUST ORDER TAKING

REPORT              SHIPPED GOODS
GENERATED-BY        SHIPPING
RECEIVED-BY         MAIL ROOM

ACTIVITY            SHIPPING
PROCESS             SHIP PRODUCT TO CUST
STIMULATED          INTERNALLY
EXECUTED-BY         SHIPPING CLERK
EXTRACTS            PACKED PRODUCT
EXTRACTS            SHIPPING NOTICE FILE
GENERATES           SHIPPING NOTICE FILE
GENERATES           SHIPPED GOODS
INCOMPATIBLE-WITH   PURCHASING
INCOMPATIBLE-WITH   RECEIVING
```

```
INCOMPATIBLE-WITH    CUST INVOICE APPROVA
INCOMPATIBLE-WITH    CUST ORDER PROCESSIN
INCOMPATIBLE-WITH    SHIPPING NOTICE PREP
INCOMPATIBLE-WITH    PACK PRODUCT
PERFORMED-BY         SHIPPING DEPT .
TRIGGERED-BY         SHIPPING NOTICE PREP
TRIGGERED-BY         PACK PRODUCT

POSITION             SHIPPING CLERK
EMPLOYED-BY          SHIPPING DEPT
EXECUTES             SHIPPING

DEPARTMENT           SHIPPING DEPT
EMPLOYS              SHIPPING CLERK
PERFORMS             SHIPPING
PREPARES             TIME CARD
RECEIVES             PACKED PRODUCT
RECEIVES             SHIPPING NOTICE FILE

REPORT               SHIPPING NOTICE FILE
INITIATED            EXTERNALLY
EXTRACTED-BY         SHIPPING
GENERATED-BY         SHIPPING NOTICE PREP
GENERATED-BY         SHIPPING
RECEIVED-BY          SHIPPING DEPT

ACTIVITY             SHIPPING NOTICE PREP
STIMULATED           INTERNALLY
EXECUTED-BY          SALES CLERK 2
EXTRACTS             CUST ORDER
GENERATES            SHIPPING NOTICE FILE
INCOMPATIBLE-WITH    SHIPPING
PERFORMED-BY         SALES DEPT
TRIGGERED-BY         CHECK QUAN ON HAND
TRIGGERS             SHIPPING

REPORT               TIME CARD
INITIATED            INTERNALLY
EXTRACTED-BY         PAYCHECK PREP
PREPARED-BY          SALES DEPT
PREPARED-BY          MAIL ROOM
PREPARED-BY          PERSONNEL DEPT
PREPARED-BY          PRODUCTION DEPT
PREPARED-BY          PURCHASING DEPT
PREPARED-BY          RECEIVING DEPT
PREPARED-BY          WAREHOUSE
PREPARED-BY          SHIPPING DEPT
PREPARED-BY          ACCOUNTING DEPT
PREPARED-BY          CREDIT DEPT
RECEIVED-BY          ACCOUNTING DEPT

REPORT               UNLOCATED ACCOUNT MS
INITIATED            EXTERNALLY
EXTRACTED-BY         MAILING TO CUST
GENERATED-BY         CASH RECEIPTS
RECEIVED-BY          MAIL ROOM

ACTIVITY             UPDATE CUST INFO
STIMULATED           INTERNALLY
EXECUTED-BY          CREDIT CLERK
EXTRACTS             CUST FILE
EXTRACTS             CUST FILE CHANGES
GENERATES            CUST FILE
PERFORMED-BY         CREDIT DEPT
TRIGGERED-BY         DETERMINE COST CHANG

ACTIVITY             UPDATE SALES FILE
STIMULATED           INTERNALLY
EXECUTED-BY          SALES CLERK 2
EXTRACTS             CUST ORDER
GENERATES            SALES FILE
PERFORMED-BY         SALES DEPT
TRIGGERED-BY         CHECK QUAN ON HAND

DEPARTMENT           VENDOR
PREPARES             PURCHASE INVOICE
PREPARES             MAIL FROM VENDOR
RECEIVES             MAIL TO VENDOR

REPORT               VENDOR CHECK
EXTRACTED-BY         MAILING TO VENDOR
GENERATED-BY         VENDOR PAYMENTS
RECEIVED-BY          MAIL ROOM

REPORT               VENDOR FILE
EXTRACTED-BY         RECEIVING
EXTRACTED-BY         PURCHASING
EXTRACTED-BY         A/P REPORTING
GENERATED-BY         PURCHASE INVOICE VER
RECEIVED-BY          PURCHASING DEPT
RECEIVED-BY          ACCOUNTING DEPT
RECEIVED-BY          RECEIVING DEPT

ACTIVITY             VENDOR PAYMENTS
STIMULATED           INTERNALLY
EXECUTED-BY          A/P CLERK
EXTRACTS             PAYMENT AUTH
GENERATES            VENDOR CHECK
INCOMPATIBLE-WITH    PARTS REQUISITIONING
INCOMPATIBLE-WITH    PURCHASE INVOICE VER
INCOMPATIBLE-WITH    MAILING TO VENDOR
INCOMPATIBLE-WITH    PAYMENT AUTH PREP
INCOMPATIBLE-WITH    MAILING FROM VENDOR
PERFORMED-BY         ACCOUNTING DEPT
TRIGGERED-BY         PAYMENT AUTH PREP
TRIGGERED-BY         PURCHASE INVOICE VER
TRIGGERS             MAILING TO VENDOR

DEPARTMENT           WAREHOUSE
EMPLOYS              WAREHOUSE CLERK
PERFORMS             PACK PRODUCT
PERFORMS             PARTS PICKING
PREPARES             TIME CARD
RECEIVES             PARTS PICKING TICKET
RECEIVES             FIN GOODS INVEN FILE
RECEIVES             PARTS INVEN FILE
RECEIVES             PACKING SLIP

POSITION             WAREHOUSE CLERK
EMPLOYED-BY          WAREHOUSE
EXECUTES             PACK PRODUCT
EXECUTES             PARTS PICKING
```

```
ACTIVITY             WAREHOUSE REPORTING
PERIOD               WEEKLY
PROCESS              REPORT VALUE AND QUA
STIMULATED           INTERNALLY
EXECUTED-BY          REPORT CLERK
EXTRACTS             PURCHASE ORDER FILE
EXTRACTS             FIN GOODS INVEN FILE
EXTRACTS             PARTS INVEN FILE
EXTRACTS             ANNUAL INVEN REPORT
GENERATES            INVEN STATUS REPORT
PERFORMED-BY         ACCOUNTING DEPT
TRIGGERED-BY         ANNUAL INVEN
TRIGGERED-BY         ACCOUNTING DEPT REQU
```

The data base has been loaded, and its contents have been dumped. A DBMS utility program summarizes the physical attributes of the data base. The results of this program (DBSM) appear on the next page.

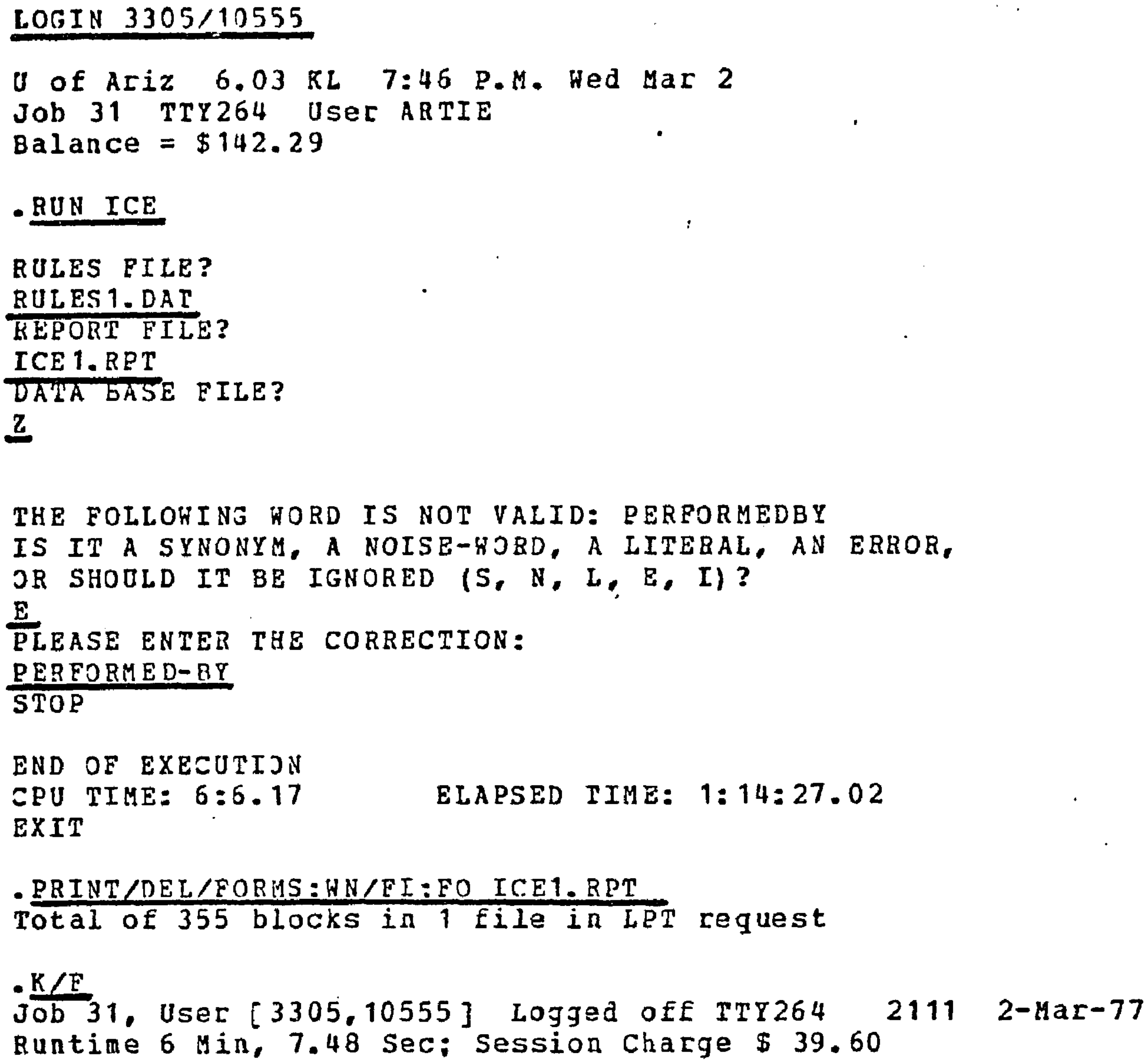
The data base occupies 31 pages (512 words each). It consists of 134 objects, 97 attributes, and 404 bidirectional relations.

```
LOGIN 3305/10555

U of Ariz  6.02 KL  1:31 P.M. Thur Mar 3
Job 12  TTY247  User ARTIE
Balance = $62.54

.RUN STDB:DBSM[2040,10103]

ENTER DATA BASE NAME
Z
ENTER DBSM REPORT FILE SPECIFICATION
TTY:

                         DBSM            13:32:21    3-Mar-77

 PAGE  MHS      NHOL  THOL  %HOL      NREC  TREC  %REC
 ----  ----     ----  ----  ----      ----  ----  ----
    1    6         1     6   1.2        26   504  98.4
    2    3         1     3   0.6        43   507  99.0
    3    3         1     3   0.6        38   507  99.0
    4    6         2    10   2.0        50   500  97.7
    5    5         1     5   1.0        47   505  98.6
    6    0         0     0   0.0        49   510  99.6
    7    0         0     0   0.0        56   510  99.6
    8    3         1     3   0.6        44   507  99.0
    9    3         1     3   0.6        52   507  99.0
   10    5         1     5   1.0        50   505  98.6
   11    6         2     9   1.8        51   501  97.9
   12    0         0     0   0.0        46   510  99.6
   13    3         1     3   0.6        54   507  99.0
   14    4         1     4   0.8        46   506  98.8
   15    0         0     0   0.0        53   510  99.6
   16    5         2     9   1.8        33   501  97.9
   17    3         2     6   1.2        44   504  98.4
   18    0         0     0   0.0        51   510  99.6
   19    3         1     3   0.6        50   507  99.0
   20    3         1     3   0.6        52   507  99.0
   21    0         0     0   0.0        53   510  99.6
   22    0         0     0   0.0        49   510  99.6
   23    0         0     0   0.0        56   510  99.6
   24    3         1     3   0.6        53   507  99.0
   25    0         0     0   0.0        55   510  99.6
   26    0         0     0   0.0        57   510  99.6
   27    0         0     0   0.0        59   510  99.6
   28    3         1     3   0.6        52   507  99.0
   29    0         0     0   0.0        64   510  99.6
   30    3         1     3   0.6        61   507  99.0
   31  149         1   149  29.1        47   361  70.5
   32 THRU    35 ARE EMPTY
 PAGE SIZE =   512 WORDS.
 NUMBER OF PAGES IN DATA BASE =   35
 TOTAL SIZE OF DATA BASE =      17920 WORDS.

 TOTAL NUMBER OF HOLES       =        27
 TOTAL HOLE SIZE             =      2273
 TOTAL PERCENT OF HOLES      =        12.68

 TOTAL NUMBER OF RECORDS     =      1541
 TOTAL RECORDS SIZE          =     15577
 TOTAL PERCENT OF RECORDS    =        86.93

 THE ABOVE SIZES ARE IN WORDS.

 RECORD PTR DATA SIZE COUNT    TOTAL   %REC   %DB
 ------ --- ---- ---- ----- -------- ------ -----
 SYSTEM   6    0    7     1        7   0.04  0.04
 OBJECT  15    8   24   134     3216  20.65 17.95
 ATTRIB   3    8   12    97     1164   7.47  6.50
 RELATE   6    4   11   501     5511  35.38 30.75
 NUB      6    0    7   808     5656  36.31 31.56

STOP

END OF EXECUTION
CPU TIME: 1.90  ELAPSED TIME: 17.38
EXIT

.K/F
Job 12, User [3305,10555]  Logged off TTY247    1334   3-Mar-77
Runtime 2.52 Sec; Session Charge $ 0.30
```

On the following page are the rules which are to be processed against the model. Interpretation of the rules appear in Section 5.3.

```
@ ACTIVITY ∃ 1 S¯(ACTIVITY EXECUTED-BY,
                ACTIVITY PERPORMED-BY) $
@ ACTIVITY ∃
      S≠ (ACTIVITY AUTHORIZED-BY POSITION [PERSON],
          ACTIVITY EXECUTED-BY POSITION [PERSON],
          ACTIVITY RECORDED-BY POSITION [PERSON]) $
@ ACTIVITY EXECUTED-BY POSITION.1 ∧ ACTIVITY RECORDED-BY POSITION.2
      ∄   DEPARTMENT EMPLOYS POSITION.1
        ∧ DEPARTMENT EMPLOYS POSITION.2 $
@ ACTIVITY.1 INCOMPATIBLE-WITH ACTIVITY.2 ∄
      SV (ACTIVITY.1 AUTHORIZED-BY POSITION AUTHORIZES ACTIVITY.2,
          ACTIVITY.1 AUTHORIZED-BY POSITION EXECUTES ACTIVITY.2,
          ACTIVITY.1 AUTHORIZED-BY POSITION RECORDS ACTIVITY.2,
          ACTIVITY.1 EXECUTED-BY POSITION AUTHORIZES ACTIVITY.2,
          ACTIVITY.1 EXECUTED-BY POSITION EXECUTES ACTIVITY.2,
          ACTIVITY.1 EXECUTED-BY POSITION RECORDS ACTIVITY.2,
          ACTIVITY.1 RECORDED-BY POSITION AUTHORIZES ACTIVITY.2,
          ACTIVITY.1 RECORDED-BY POSITION EXECUTES ACTIVITY.2,
          ACTIVITY.1 RECORDED-BY POSITION RECORDS ACTIVITY.2)  $
@ POSITION.1 ∄
      (POSITION.1 (MANAGES POSITION.2)! MANAGES POSITION.1 $
@ ACTIVITY.1 (TRIGGERS ACTIVITY.2)! TRIGGERS ACTIVITY.3 ∃
      S≠ (ACTIVITY.1 AUTHORIZED-BY [PERSON],
          ACTIVITY.3 EXECUTED-BY [PERSON],
          ACTIVITY.3 RECORDED-BY [PERSON]) $
@ POSITION.1 ∄
          (POSITION.1 (MANAGES POSITION.2)! MANAGES POSITION.1) $
@ POSITION ∃ 1 POSITION MANAGED-BY $
@ POSITION ∃ SV(POSITION AUTHORIZES,
               POSITION EXECUTES,
               POSITION RECORDS)   $
@ POSITION ∃ POSITION EMPLOYED-BY$
@ POSITION ∃ POSITION PERSON $
@ ACTIVITY ∃ S¯(ACTIVITY EXECUTED-BY,
               ACTIVITY TRIGGERS,
               ACTIVITY TRIGGERED-BY,
               ACTIVITY PREFORMED-BY,
               ACTIVITY GENERATES,
               ACTIVITY EXTRACTS) $
@ ACTIVITY [STIMULATED] ∈ ('INTERNALLY', 'EXTERNALLY')$
@ ACTIVITY [STIMULATED] = 'EXTERNALLY'
    ∄ ACTIVITY TRIGGERED-BY $
@ ACTIVITY [STIMULATED] = 'INTERNALLY'
    ∃ ACTIVITY TRIGGERED-BY $

@ DEPARTMENT RECEIVES REPORT ∃
    DEPARTMENT PERFORMS ACTIVITY EXTRACTS REPORT $
@ ACCOUNT ∃ [ACCOUNT-TYPE] ≠ '    ' $
@ ACCOUNT [ACCOUNT-TYPE] ∈ ('ASSET', 'LIABILITY', 'EQUITY') $
@ ACCOUNT ∃ ACCOUNT DEBITED-BY ∧ ACCOUNT CREDITED-BY$
@ ACTIVITY.1 DEBITS ∃
      (ACTIVITY.1 (TRIGGERS ACTIVITY.2)! CREDITS) V
      (ACTIVITY.1 (TRIGGERED-BY ACTIVITY.2)! CREDITS) $
@ ACTIVITY.1 CREDITS∃
      (ACTIVITY.1 (TRIGGERS ACTIVITY.2)! DEBITS) V
      (ACTIVITY.1 (TRIGGERED-BY ACTIVITY.2)! DEBITS) $
@ REPORT ∃ S¯(REPORT GENERATED-BY,
             REPORT EXTRACTED-BY,
             REPORT PREPARED-BY,
             REPORT RECEIVED-BY) $
```

Following are some of the interactions and outputs from processing the rules against the data base. All of the rules were processed in three runs. They required over seven minutes of execution time, one hour 31 minutes of "wall clock" time, and $46.42.



```
LOGIN 3305/10555

U of Ariz  6.03 KL  7:46 P.M. Wed Mar 2
Job 31  TTY264  User ARTIE
Balance = $142.29

.RUN ICE

RULES FILE?
RULES1.DAT
REPORT FILE?
ICE1.RPT
DATA BASE FILE?
Z


THE FOLLOWING WORD IS NOT VALID: PERFORMEDBY
IS IT A SYNONYM, A NOISE-WORD, A LITERAL, AN ERROR,
OR SHOULD IT BE IGNORED (S, N, L, E, I)?
E
PLEASE ENTER THE CORRECTION:
PERFORMED-BY
STOP

END OF EXECUTION
CPU TIME: 6:6.17        ELAPSED TIME: 1:14:27.02
EXIT

.PRINT/DEL/FORMS:WN/FI:FO ICE1.RPT
Total of 355 blocks in 1 file in LPT request

.K/F
Job 31, User [3305,10555] Logged off TTY264   2111  2-Mar-77
Runtime 6 Min, 7.48 Sec; Session Charge $ 39.60
```

The first rule contained a typographical error. PERFORMEDBY was missing a hyphen, and the ICE scanner requested its correction. This interaction took place at the terminal (as appears on the preceding page). The output report lists the rule as it was inputed, and then lists the rule's violations.

```
@ ACTIVITY ∃ 1 S^(ACTIVITY EXECUTED-BY,
                ACTIVITY PERFORMEDBY) $


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ..........................................
    :ACTIVITY           . :HANDLE PARTS DISCREP:


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ..........................................
    :ACTIVITY             :ACCOUNTING DEPT REQU:


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ..........................................
    :ACTIVITY             :SALES DEPT REQUEST  :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ..........................................
    :ACTIVITY             :PURCHASING DEPT REQU:


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ..........................................
    :ACTIVITY             :PRODUCTION DEPT REQU:


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ..........................................
    :ACTIVITY             :PERSONNEL DEPT REQUE:


*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

The preceding rule is a completeness check. The ACTIVITY, HANDLE PARTS DISCREPANCY, is not executed by any position (this is noted by referencing the formatted listing). This is an incompleteness that should be corrected.

The five department requests were intentionally defined incompletely. These are activities which have been defined for causing a certain class of standard reports to be processed.

```
∂ ACTIVITY.1 INCOMPATIBLE-WITH ACTIVITY.2 ∄
        SV (ACTIVITY.1 AUTHORIZED-BY POSITION AUTHORIZES ACTIVITY.2,
            ACTIVITY.1 AUTHORIZED-BY POSITICN EXECUTES ACTIVITY.2,
            ACTIVITY.1 AUTHORIZED-BY POSITION RECORDS ACTIVITY.2,
            ACTIVITY.1 EXECUTED-BY POSITION AUTHORIZES ACTIVITY.2,
            ACTIVITY.1 EXECUTED-BY POSITION EXECUTES ACTIVITY.2,
            ACTIVITY.1 EXECUTED-BY POSITION RECORDS ACTIVITY.2,
            ACTIVITY.1 RECORDED-BY POSITION AUTHORIZES ACTIVITY.2,
            ACTIVITY.1 RECORDED-BY POSITION EXECUTES ACTIVITY.2,
            ACTIVITY.1 RECORDED-BY POSITION RECORDS ACTIVITY.2)   $


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ...........................................
    :ACTIVITY             :MAILING TO VENDOR   :
    :INCOMPATIBLE-WITH    :
    :ACTIVITY             :MAILING FROM VENDOR :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ...........................................
    :ACTIVITY             :MAILING TO CUST     :
    :INCOMPATIBLE-WITH    :
    :ACTIVITY             :MAILING FROM CUST   :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOJING OCCURRENCE:
    D B NAME              : VALUE
    ...........................................
    :ACTIVITY             :MAILING FROM CUST   :
    :INCOMPATIBLE-WITH    :
    :ACTIVITY             :MAILING TO CUST     :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ...........................................
    :ACTIVITY             :MAILING FROM VENDOR :
    :INCOMPATIBLE-WITH    :
    :ACTIVITY             :MAILING TO VENDOR   :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ...........................................
    :ACTIVITY             :CUST ORDER PROCESSIN:
    :INCOMPATIBLE-WITH    :
    :ACTIVITY             :CHECK CREDIT CEILING:


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ...........................................
    :ACTIVITY             :CUST ORDER PROCESSIN:
    :INCOMPATIBLE-WITH    :
    :ACTIVITY             :CHECK CREDIT RATING :
```

```
A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :ACTIVITY            :PACK PRODUCT        :
    :INCOMPATIBLE-WITH   :
    :ACTIVITY            :PARTS PICKING       :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :ACTIVITY            :VENDOR PAYMENTS     :
    :INCOMPATIBLE-WITH   :
    :ACTIVITY            :PURCHASE INVOICE VER:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :ACTIVITY            :PURCHASE INVOICE VER:
    :INCOMPATIBLE-WITH   :
    :ACTIVITY            :VENDOR PAYMENTS     :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :ACTIVITY            :PARTS PICKING       :
    :INCOMPATIBLE-WITH   :
    :ACTIVITY            :PACK PRODUCT        :

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

Incompatible functions with undesirable relations have been found by the preceding rule.

```
@ POSITION.1 ]̸
        (POSITION.1 (MANAGES POSITION.2)! MANAGES POSITION.1 $

***** MISSING ) *****

*****SYNTAX ERROR IN PARSE: $                          *****
```

The preceding rule contains a syntax error and was not processed. A later run reprocesses it correctly.

```
@ POSITION ∃ 1 POSITION MANAGED-BY $


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    .........................................
    :POSITION                :MAIL CLERK          :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    .........................................
    :POSITION                :REPORT CLERK        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    .........................................
    :POSITION                :PURCHASING CLERK    :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    .........................................
    :POSITION                :SHIPPING CLERK      :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    .........................................
    :POSITION                :PERSONNEL CLERK     :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    .........................................
    :POSITION                :ACCOUNTANT 1        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    .........................................
    :POSITION                :SALES CLERK 3       :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    .........................................
    :POSITION                :PAYROLL CLERK       :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    .........................................
    :POSITION                :MACHINE OPERATOR    :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    ..........................................
    :POSITION            :WAREHOUSE CLERK     :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ..........................................
    :POSITION            :PRODUCTION CLERK    :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ..........................................
    :POSITION            :A/P CLERK           :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ..........................................
    :POSITION            :RECEIVING CLERK 2   :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ..........................................
    :POSITION            :RECEIVING CLERK 1   :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ..........................................
    :POSITION            :A/R CLERK           :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ..........................................
    :POSITION            :SALES CLERK 2       :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ..........................................
    :POSITION            :CREDIT CLERK        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ..........................................
    :POSITION            :SALES CLERK 1       :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ..........................................
    :POSITION            :SALESPERSON         :

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

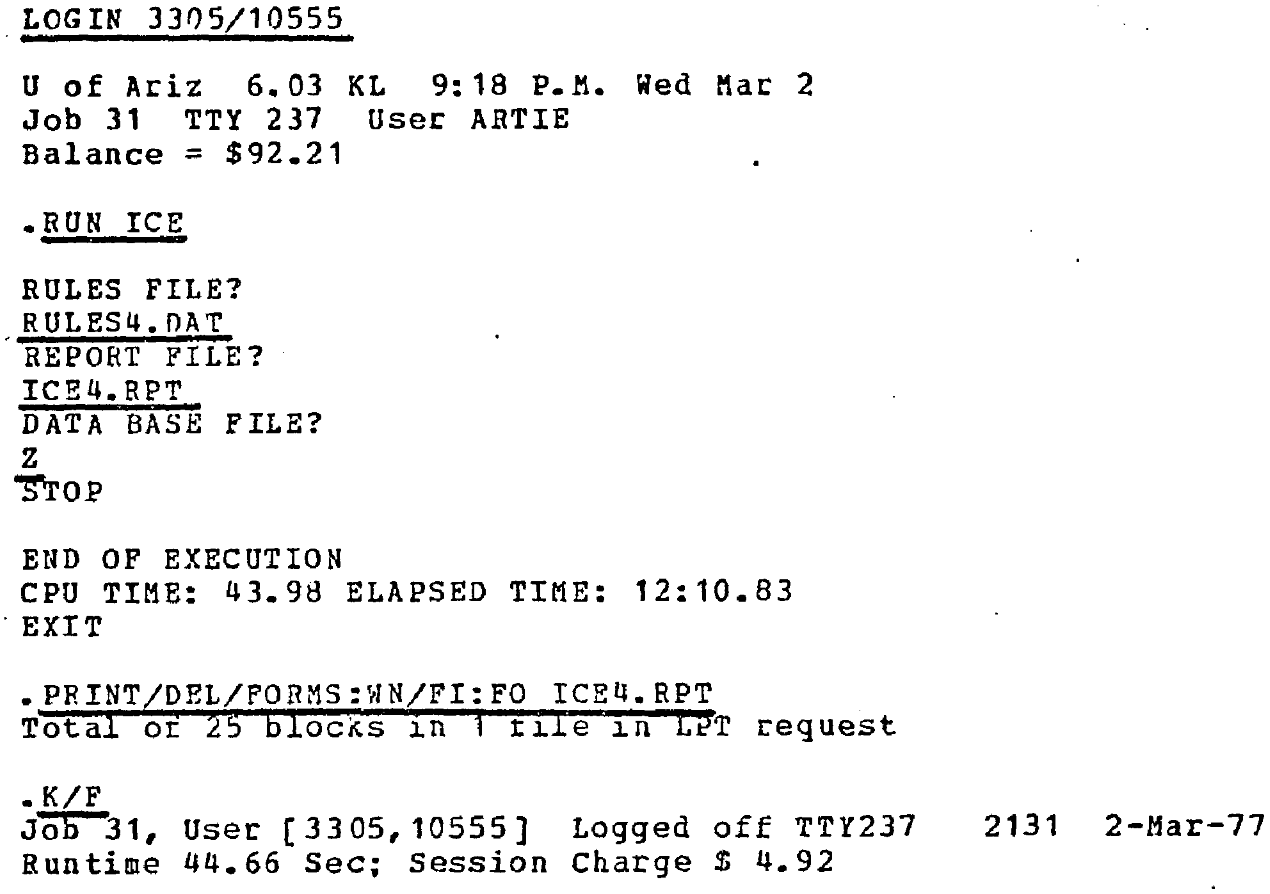
The preceding violations indicate that the organizational chart has not been defined.

```
∂ ACTIVITY [STIMULATED] ∈ ('INTERNALLY', 'EXTERNALLY')$

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

```
∂ ACTIVITY [STIMULATED] = 'EXTERNALLY'
      ∄ ACTIVITY TRIGGERED-BY $

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

```
@ ACTIVITY [STIMULATED] = 'INTERNALLY'
     ∃ ACTIVITY TRIGGERED-BY $


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ........................................
    :ACTIVITY             :ACCOUNTING DEPT REQU:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ........................................
    :ACTIVITY             :SALES DEPT REQUEST  :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ........................................
    :ACTIVITY             :PURCHASING DEPT REQU:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ........................................
    :ACTIVITY             :PRODUCTION DEPT REQU:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME              : VALUE
    ........................................
    :ACTIVITY             :PERSONNEL DEPT REQUE:

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

The preceding rule yielded expected violations.



```
LOGIN 3305/10555

U of Ariz  6.03 KL  9:18 P.M. Wed Mar 2
Job 31  TTY 237  User ARTIE
Balance = $92.21

.RUN ICE

RULES FILE?
RULES4.DAT
REPORT FILE?
ICE4.RPT
DATA BASE FILE?
Z
STOP

END OF EXECUTION
CPU TIME: 43.98 ELAPSED TIME: 12:10.83
EXIT

.PRINT/DEL/FORMS:WN/FI:FO ICE4.RPT
Total of 25 blocks in 1 file in LPT request

.K/F
Job 31, User [3305,10555]  Logged off TTY237   2131  2-Mar-77
Runtime 44.66 Sec; Session Charge $ 4.92
```

```
∂ REPORT ∃ S¯(REPORT GENERATED-BY,
              REPORT EXTRACTED-BY,
              REPORT PREPARED-BY,
              REPORT RECEIVED-BY) $ .


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :CUST ORDER          :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :CUST BACKORDER MSG#1:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :SHIPPED GOODS       :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :EMPLOYEE CHANGE FORM:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :PACKED PRODUCT      :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :ANNUAL INVEN REPORT :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :OPERATING INFO FILE :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :RECORD OF EARNINGS  :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :EMPLOYEE FILE       :
```

```
A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :PAYCHECK            :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :EMPLOYEE TAX REPORT :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :COMPANY TAX REPORT  :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :A/R REPORT          :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :A/P REPORT          :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :INVEN STATUS REPORT :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :PROFIT ANALYSIS REPO:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :CUST BACKORDER REPOR:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :CUST REPORT         :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :SALES REPORT        :
```

```
A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :PRODUCTION REPORT   :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :TIME CARD           :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :PICKED PARTS        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :PURCHASE ORDER FILE :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :PARTS INVEN FILE    :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :PURCHASE ORDER #2   :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :VENDOR FILE         :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :ACCOUNT BALANCE NEG :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :SALES FILE          :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME                 : VALUE
    •••••••••••••••••••••••••••••••••••••••••••
    :REPORT              :CUST BACKORDER FILE :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :FIN GOODS INVEN FILE:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :CUST FILE           :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :MAIL FROM VENDOR    :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :MAIL FROM CUST      :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :MAIL TO CUST        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :PURCHASE ORDER #1   :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :VENDOR CHECK        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :MAIL TO VENDOR      :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :UNLOCATED ACCOUNT MS:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :PARTS RECEIPT DISCRE:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :PAYMENT AUTH        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :PARTS PICKING TICKET:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :PACKING SLIP        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :SHIPPING NOTICE FILE:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :CREDIT CEILING EXCEE:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :CUST INVOICE        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :NEW CUST MSG        :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :RECEIVING REPORT    :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :BAD CREDIT RATING MS:

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :REPORT              :CUST BACKORDER MSG#2:

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

For most of the violations of the preceding rule, the relation PERFORMED-BY was not defined. To determine which relations were not defined for the reports it would be necessary to reference the formatted listing or to run four rules (each specifying one type of relation).

```
LOGIN 3305/10555

U of Ariz  6.03 KL  9:04 P.M. Thur Mar 3
Job 55  TTY262  User ARTIE
Balance = $53.14

.RUN ICE

RULES FILE?
RULES5.DAT
REPORT FILE?
ICE5.RPT
DATA BASE FILE?
Z
STOP

END OF EXECUTION
CPU TIME: 3.05  ELAPSED TIME: 33.83
EXIT

.PRINT/DEL/FORMS:NN/FI:FO ICE5.RPT
Total of 1 block in 1 file in LPT request

.K/F
Job 55, User [3305,10555]  Logged off TTY262   2105  3-Mar-77
Runtime 3.96 Sec; Session Charge $ 0.47
```

```
LIST POSITION$

    :POSITION            :MAIL CLERK            :

    :POSITION            :REPORT CLERK          :

    :POSITION            :PURCHASING CLERK      :

    :POSITION            :SHIPPING CLERK        :

    :POSITION            :PERSONNEL CLERK       :

    :POSITION            :ACCOUNTANT 1          :

    :POSITION            :SALES CLERK 3         :

    :POSITION            :PAYROLL CLERK         :

    :POSITION            :MACHINE OPERATOR      :

    :POSITION            :WAREHOUSE CLERK       :

    :POSITION            :PRODUCTION CLERK      :

    :POSITION            :A/P CLERK             :

    :POSITION            :RECEIVING CLERK 2     :

    :POSITION            :RECEIVING CLERK 1     :

    :POSITION            :A/R CLERK             :

    :POSITION            :SALES CLERK 2         :

    :POSITION            :CREDIT CLERK          :

    :POSITION            :SALES CLERK 1         :

    :POSITION            :SALESPERSON           :

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

```
          LIST POSITION MANAGES$

          THERE ARE NO OCCURRENCES IN THE DATA BASE.

          *****END OF COMPLIANCE TEST FOR THIS RULE*****
```

```
∂ POSITION.1 ∄
        (POSITION.1 (MANAGES POSITION.2)! MANAGES POSITION.1) $


*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

It is necessary at this time to load additional data into the data base to define the organizational chart.

The interactions on the next page show that a copy of the data base is created before updating it. If the computer system were to "go down" during the execution of the job, or if the job were interrupted in any other manner, the data base would be unrecoverable. After the job has properly halted, the backup is deleted.

```
LOGIN 3305/10555

U of Ariz  6.02 KL  1:33 P.M. Fri Mar 4
Job 57  TTY256  User ARTIE
Balance = $40.55

.COPY DB.BAK=Z.DB

.RUN LOADER

DATA FILE?
Z.2
REPORT FILE?
LOADER.RPT
DATA BASE FILE?
Z
* END OF JOB *
STOP

END OF EXECUTION
CPU TIME: 1:59.19        ELAPSED TIME: 1:23:57.48
EXIT

.DEL DB.BAK
Files deleted from STDN:[3305,10555]
DB.BAK
144 blocks freed

.K/F
Other jobs same PPN
Job 57, User [3305,10555] Logged off TTY256   1458  4-Mar-77
Runtime 2 Min, 0.06 Sec; Session Charge $ 13.92
```

The high cost of the run is quickly noticed, considering that only 22 relations and six new objects were loaded (as appears on the following page). The data base is already very large, and much searching is required to load each relation. Recall that: 1) both object-values of the relations are searched to assure that the relation is consistent; 2) the relation is searched to see if it has already been defined for the values; and 3) the relation requires two loads, one for each direction of the relationship. Although the data base sets are sorted, the DBMS search routine is very inefficient, using a sequential search algorithm.

```
        ***** ORGANIZATIONAL CHART *****

:POSITION            :PRESIDENT
:MANAGES             :VP PRODUCTION
:MANAGES             :VP SALES
:MANAGES             :VP FINANCE

:POSITION            :VP PRODUCTION
:MANAGES             :MACHINE OPERATOR
:MANAGES             :PRODUCTION CLERK
:MANAGES             :PURCH & RECEIVING MG
:MANAGED-BY          :WAREHOUSE CLERK

:POSITION            :PURCH & RECEIVING MG
:MANAGES             :RECEIVING CLERK 1
:MANAGES             :RECEIVING CLERK 2
:MANAGES             :WAREHOUSE CLERK

:POSITION            :VP SALES
:MANAGES             :SHIPPING CLERK
:MANAGES             :SALES MANAGER

:POSITION            :SALES MANAGER
:MANAGES             :SALES DEPT
*****THE OPERAND HAS ALREADY BEEN DEFINED AS A DEPARTMENT
*****AND CANNOT BE REDEFINED - IGNORED.
:MANAGES             :SALESPERSON
:MANAGES             :SALES CLERK 1
:MANAGES             :SALES CLERK 2
:MANAGES             :SALES CLERK 3

:POSITION            :SALES CLERK 3
:MANAGES             :SHIPPING CLERK

:POSITION            :VP FINANCE
:MANAGES             :ACCOUNTANT 1
:MANAGES             :A/P CLERK
:MANAGES             :A/R CLERK
:MANAGES             :CREDIT CLERK
```

The preceding page is the output from the loader program that copies the input and lists error messages. The loader found an error in the use of the MANAGES relation. The operands of MANAGES must be positions, but SALES DEPT had been defined earlier as a department. An allowable relation would have been EMPLOYED-BY.

After loading the organizational chart, it becomes necessary to reprocess those rules that are affected by the plan of organization. The interactions involving some of those rules follow.

```
LOGIN 3305/10555

U of Ariz  6.02 KL  3:11 P.M. Fri Mar 4
Job 44  TTY260  User ARTIE
Balance = $25.64

.RUN ICE

RULES FILE?
RULES4.DAT
REPORT FILE?
ICE.RPT
DATA BASE FILE?
Z
STOP

END OF EXECUTION
CPU TIME: 12.93 ELAPSED TIME: 12:21.45
EXIT

.PRINT/DEL/FORMS:WN/FI:FO ICE.RPT
Total of 7 blocks in 1 file in LPT request

.K/F
Job 44, User [3305,10555]  Logged off TTY260   1525  4-Mar-77
Runtime 13.59 Sec; Session Charge $ 1.65
```

```
LIST POSITION $

    :POSITION             :MAIL CLERK             :
    :POSITION             :REPORT CLERK           :
    :POSITION             :PURCHASING CLERK       :
    :POSITION             :SHIPPING CLERK         :
    :POSITION             :PERSONNEL CLERK        :
    :POSITION             :ACCOUNTANT 1           :
    :POSITION             :SALES CLERK 3          :
    :POSITION             :PAYROLL CLERK          :
    :POSITION             :MACHINE OPERATOR       :
    :POSITION             :WAREHOUSE CLERK        :
    :POSITION             :PRODUCTION CLERK       :
    :POSITION             :A/P CLERK              :
    :POSITION             :RECEIVING CLERK 2      :
    :POSITION             :RECEIVING CLERK 1      :
    :POSITION             :A/R CLERK              :
    :POSITION             :SALES CLERK 2          :
    :POSITION             :CREDIT CLERK           :
    :POSITION             :SALES CLERK 1          :
    :POSITION             :SALESPERSON            :
    :POSITION             :PRESIDENT              :
    :POSITION             :VP FINANCE             :
    :POSITION             :VP SALES               :
    :POSITION             :SALES MANAGER          :
    :POSITION             :VP PRODUCTION          :
    :POSITION             :PURCH & RECEIVING MG:

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

```
LIST POSITION.1 MANAGED-BY POSITION.2 $

     :POSITION              :SHIPPING CLERK      :
     :MANAGED-BY            :
     :POSITION              :VP SALES            :

     :POSITION              :SHIPPING CLERK      :
     :MANAGED-BY            :
     :POSITION              :SALES CLERK 3       :

     :POSITION              :ACCOUNTANT 1        :
     :MANAGED-BY            :
     :POSITION              :VP FINANCE          :

     :POSITION              :SALES CLERK 3       :
     :MANAGED-BY            :
     :POSITION              :SALES MANAGER       :

     :POSITION              :MACHINE OPERATOR    :
     :MANAGED-BY            :
     :POSITION              :VP PRODUCTION       :

     :POSITION              :WAREHOUSE CLERK     :
     :MANAGED-BY            :
     :POSITION              :PURCH & RECEIVING MG:

     :POSITION              :PRODUCTION CLERK    :
     :MANAGED-BY            :
     :POSITION              :VP PRODUCTION       :

     :POSITION              :A/P CLERK           :
     :MANAGED-BY            :
     :POSITION              :VP FINANCE          :

     :POSITION              :RECEIVING CLERK 2   :
     :MANAGED-BY            :
     :POSITION              :PURCH & RECEIVING MG:

     :POSITION              :RECEIVING CLERK 1   :
     :MANAGED-BY            :
     :POSITION              :PURCH & RECEIVING MG:

     :POSITION              :A/R CLERK           :
     :MANAGED-BY            :
     :POSITION              :VP FINANCE          :

     :POSITION              :SALES CLERK 2       :
     :MANAGED-BY            :
     :POSITION              :SALES MANAGER       :

     :POSITION              :CREDIT CLERK        :
     :MANAGED-BY            :
     :POSITION              :VP FINANCE          :

     :POSITION              :SALES CLERK 1       :
     :MANAGED-BY            :
     :POSITION              :SALES MANAGER       :

     :POSITION              :SALESPERSON         :
```

```
    :MANAGED-BY           :
    :POSITION             :SALES MANAGER       :

    :POSITION             :VP FINANCE          :
    :MANAGED-BY           :
    :POSITION             :PRESIDENT           :

    :POSITION             :VP SALES            :
    :MANAGED-BY           :
    :POSITION             :PRESIDENT           :

    :POSITION             :SALES MANAGER       :
    :MANAGED-BY           :
    :POSITION             :VP SALES            :

    :POSITION             :VP PRODUCTION       :
    :MANAGED-BY           :
    :POSITION             :PRESIDENT           :

    :POSITION             :VP PRODUCTION       :
    :MANAGED-BY           :
    :POSITION             :WAREHOUSE CLERK     :

    :POSITION             :PURCH & RECEIVING MG:
    :MANAGED-BY           :
    :POSITION             :VP PRODUCTION       :

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

```
∂ POSITION.1 ∃ 1 POSITION.1 MANAGED-BY POSITION.2 $


A VIOLATION. HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :POSITION            :MAIL CLERK           :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :POSITION            :REPORT CLERK         :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :POSITION            :PURCHASING CLERK     :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :POSITION            :SHIPPING CLERK       :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :POSITION            :PERSONNEL CLERK      :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :POSITION            :PAYROLL CLERK        :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :POSITION            :PRESIDENT            :


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
    D B NAME             : VALUE
    ...........................................
    :POSITION            :VP PRODUCTION        :


*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

A violation of the preceding rule indicates that a position has more than one manager or it has no manager. The mail clerk, report clerk, personnel clerk, purchasing clerk, and payroll clerk do not have defined managers; the MANAGED-BY relations should be defined later for these positions. The shipping clerk has two managers; this will be corrected in a later run. The president has no manager; this is an expected violation.

```
@ POSITION.1 ∄
        (POSITION.1 (MANAGES POSITION.2)! MANAGES POSITION.1) $


A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
   D B NAME                 : VALUE
   .....................................
   :POSITION            .   :WAREHOUSE CLERK       :

A VIOLATION HAS BEEN FOUND FOR THE FOLLOWING OCCURRENCE:
   D B NAME                 : VALUE
   .....................................
   :POSITION                :VP PRODUCTION         :

*****END OF COMPLIANCE TEST FOR THIS RULE*****
```

The preceding rule found two people who indirectly manage themselves. The Warehouse Clerk manages the VP Production who, in turn, manages the Warehouse Clerk. Also, the VP Production manages the Warehouse Clerk who, in turn, manages the VP Production. This will be corrected.

There are (at least) two corrections to be made to the data base. The Shipping Clerk should not be managed by the Sales Clerk 3, and the Warehouse Clerk should not manage the VP Productions. The interactions required to delete these data follow (refer to Section 4.3.3).

```
LOGIN 3305/10555

U of Ariz  6.02 KL  3:16 P.M. Sat Mar 5
Job 20  TTY236  User ARTIE
Balance = $202.65

.RUN DELETE

DATA BASE FILE?
Z


LONER OBJECTS, RELATION, LONER NUBS, ATTRIBUTE, STOP?' (O, R, N, A, S)
R
INPUT FILE?
TTY:
REPORT FILE?
TTY:
TYPE "END" TO STOP.

OBJECT-TYPE 1?
POSITION
OBJECT VALUE 1?
SHIPPING CLERK
RELATION-TYPE?
MANAGED-BY
OBJECT-TYPE 2?
POSITION
OBJECT VALUE 2?
SALES CLERK 3
INVERSE RELATION-TYPE?
MANAGES
[DELETED]

OBJECT-TYPE 1?
POSITION
OBJECT VALUE 1?
WAREHOUSE CLERK
RELATION-TYPE?
MANAGES
OBJECT-TYPE 2?
POSITION
OBJECT VALUE 2?
V P PRODUCTION
***** NOT FOUND *****

OBJECT-TYPE 1?
POSITION
OBJECT VALUE 1?
WAREHOUSE CLERK
RELATION-TYPE?
MANAGES
OBJECT-TYPE 2?
POSITION
OBJECT VALUE 2?
VP PRODUCTION
INVERSE RELATION-TYPE?
MANAGED-BY
[DELETED]

OBJECT-TYPE 1?
```

```
END


LONER OBJECTS, RELATION, LONER NUBS, ATTRIBUTE, STOP?' (O, R, N, A, S)
S


END OF JOB
STOP

END OF EXECUTION
CPU TIME: 7.31  ELAPSED TIME: 3:1.98
EXIT

.K/F
Job 20, User [3305,10555]  Logged off TTY236   1519  5-Mar-77
Runtime 7.94 Sec; Session Charge $ 0.70
```

REFERENCES

Adams, Donald L. and John F. Mullarkey, "A Survey of Audit Software," Journal of Accountancy (September 1972), pp. 39-66.

American Institute of Certified Public Accountants (AICPA), Codification of Auditing Standards and Procedures, 1976.

Anderson, Rodney J., "Analytical Auditing -- Does It Work?", The Internal Auditor (July/August 1972), pp. 36-54.

Arens, Alvin A. and James K. Loebbecke, Auditing: An Integrated Approach, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1976.

Auditing Advanced EDP Systems Task Force, "Advanced EDP Systems and the Auditor's Concerns," The Journal of Accountancy (January 1975), pp. 66-72.

Bodnar, George, "Reliability Modeling of Internal Control Systems," The Accounting Review (October 1975), pp. 747-757.

Brown, R. Gene, "Objective Internal Control Evaluation," The Journal of Accountancy (November 1962), pp. 50-56.

Burns, David C., and James K. Loebbecke, "Internal Control Evaluation: How the Computer Can Help," The Journal of Accountancy (August 1975), pp. 60-70.

Chandler, Donald and John H. Mullin, "Solving the 'Invisible Record' Problem," The Price Waterhouse Review (Summer/Autumn 1971).

Committee on Auditing Procedure, Internal Control, New York: American Institute of Accountants, 1949.

Cushing, Barry E., "A Mathematical Approach to the Analysis and Design of Internal Control Systems," The Accounting Review (January 1974), pp. 24-41.

Hannum, William H., "A Model for Evaluating Imperfect Control Systems," Decision Sciences (Vol. 5 1974), pp. 311-322.

Hershey, Ernest A. and Paul W. Messink, "A Data Base Management System for PSA Based on DBTG 71," ISDOS Working Paper No. 88, University of Michigan, July 1975.

Horwitz, Geoff, "Needed: A Computer Audit Philosophy," The Journal of Accountancy (April 1976), pp. 69-72.

John, Richard C. and Thomas J. Nissen, "Evaluating Internal Control in EDP Audits," The Journal of Accountancy (February 1970), pp. 31-38.

Jones, Thomas, personal interview, Ernst & Ernst (June 1977).

Kehl, Eric Edward, Development of an Accounting Problem Statement Language, M.S. Thesis, Department of Management Information Systems, University of Arizona, Tucson, 1977.

Lewis, William F., "Auditing Concepts & On-Line Computer Systems," The Arthur Young Journal (Winter/Spring 1971), pp. 3-15.

Lieberman, Arthur Z., "A Methodology for Automation of the Financial Audit Process," Proceedings of the Hawaii International Conference on Systems Sciences, 1977.

Lieberman, Arthur Z., J. F. Nunamaker, and Hugh Warren, "A Language for Modeling the Financial Audit Process," Proceedings of the Sixth Annual Pittsburgh Conference on Modeling and Simulation, 1975, pp. 1281-1286.

Meigs, Walter B., E. John Larsen, and Robert F. Meigs, Principles of Auditing, Homewood, Illinois: Richard D. Irwin, Inc., 1973.

Norrid, Lawrence R., personal interview, University of Arizona (June 1977).

Nunamaker, J. F. and Benn Konsynski, Company Z: An Information Systems Design Problem, University of Arizona, 1975.

Peat, Marwick, Mitchell & Co., Audit Manual, April 1976.

Porter, W. Thomas, EDP Controls and Auditing, Belmont California: Wadsworth Publishing Company, Inc., 1974.

Rittersbach, George H. and Stephen D. Harlan, "Auditing Advanced Systems," The Journal of Accountancy (June 1974), pp. 83-85.

Stickler, A. D., "An Appraisal of Flow Charting as an Audit Technique," Canadian Chartered Accountant (December 1968), pp. 412-415.

Study Group on Computer Control and Audit Guidelines, Computer Control Guidelines, Toronto: The Canadian Institute of Chartered Accountants, 1970.

Study Group on Computer Control and Audit Guidelines, Computer Audit Guidelines, Toronto: The Canadian Institute of Chartered Accountants, 1975.

Teichroew, Daniel and Hasan Sayani, "Automation of System Building," in System Analysis Techniques, J. Daniel Couger and Rober W. Knapp, eds., New York: John Wiley & Sons, 1974, pp. 379-389.

Thierman, Elmer, personal interview, Peat, Marwick, Mitchell and Co. (June 1977).

Will, Hartmut J., "Model Management Systems," in Information Systems and Organizational Structure, edited by Erwin Grochla and Norbert Szyperski, Berlin: Walter de Gruyter & Co., 1975.

Will, Harmut J., "Auditor/ACL Interface: Design and Behaviour" (Working Paper No. 374a), University of British Columbia, 1976.

Yu, Seongjae and John Neter, "A Stochastic Model of the Internal Control System," Journal of Accounting Research (Autumn 1973), pp. 273-295.