AN ARCHITECTURE
FOR COMPUTER-BASED
ACCOUNTING INFORMATION SYSTEMS

PhD Thesis

Peter Seddon
Department of Accounting and Finance
The University of Melbourne

3 March, 1991

Acknowledgments

Thanks to my supervisors, Ken Wright, Tony Montgomery, and Bob Nicol, particularly Ken, for the time they devoted to reading and commenting on this lengthy tome and the working papers that preceded it. Thanks also to Liz and Neil Roberts for believing in me. Of all the work I read in writing this thesis, the writer with whom I felt most empathy was Sweeney [1936]. Hopefully, this thesis will help keep Sweeney's ideas alive.
TO WHOM IT MAY CONCERN

This is to certify that (i) the thesis comprises only my original work, (ii) due acknowledgement has been made in the text of the thesis to all other material used, (iii) the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies, appendices and footnotes.

Signature: Peter Seld

Please print name in full: Peter Beaufort Seppow

Department: Accounting & Finance

Date: 13 March, 1991...
AN ARCHITECTURE FOR COMPUTER-BASED ACCOUNTING INFORMATION SYSTEMS

ABSTRACT

The question addressed in this thesis is whether cost-effective, computer-based accounting systems can be used to generate "better" accounting information than existing transaction processing accounting systems.

The first half of the thesis is devoted to gathering and summarizing information about how computer-based accounting systems work today, and what might constitute "better" accounting information. Data about present-day computer-based accounting systems was collected by mail questionnaires and personal reviews of widely-used packaged accounting software. Information about what constitutes "better" accounting information was collected, first, by reviewing the normative accounting literature, second, by reviewing the empirical literature on the inflation accounting experiments in the US (SFAS 33) and UK (SSAP 16), and third, by reviewing the academic literature on computer-based accounting, particularly the work of Ijiri, McCarthy, and Weber.

To reconcile the apparent conflict between the empirical evidence that (a) inflation accounting is essential in times of very high inflation and (b) the empirical studies had found very little evidence of additional information content in SFAS 33 and SSAP 16 reports, it is suggested that the benefits of inflation accounting must only become apparent when general price-level changes exceed, say, 15%-20% p.a.. Thus the ideas of the normative theorists are not rejected, and it is decided that a computer-based general ledger system
that (a) is inflation-tolerant, (b) draws its data from the firm's transaction processing system database, and (c) can provide accounting reports based on different sets of accounting rules (called Multiview Accounting), would be likely to meet the objectives for the thesis.

The second half of the thesis focuses on the design of such a system. To build inflation-tolerance into the profit measurement system, it is proposed that the constant-value journal entries and constant-value ledger account balances of conventional ledger systems should be replaced by formulae like those in spreadsheets. It is shown that a coherent system of double-entry bookkeeping, called Formula Accounting, can be developed, where ledger account balances may be functions of any variable that is likely to change in value over time, e.g., time itself, stock market prices, and price-index series.

For automatic generation of Formula Accounting (FA) journal entries it is proposed that either the firm's many special-purpose transaction processing systems should be modified, or that a combination of (a) a specially-defined accounting data model (called the REE model), and (b) a computer program that encodes the rules used by accountants when they prepare journal entries (called an Interpreter), should be developed. To demonstrate the feasibility of all these proposals, a prototype REE-Interpreter-FA system was developed in roughly 4,000 lines of Prolog. Multiview Accounting is illustrated by using the prototype system to generate Historical Cost and Current Cost/Constant Purchasing Power interpretations of representative Exchange Events for both a trading firm and a manufacturing firm.
TABLE OF CONTENTS

(i) Standard Abbreviations

1. Introduction 3100

1.1 Statement of Objectives
1.2 Chapter structure of this thesis

PART A: REQUIREMENTS DETERMINATION

2. Normative Accounting Theory 11700

2.1 The objectives of accounting
2.2 Information required to meet the three objectives
2.3 What data to collect?
2.4 Description of the major accounting system proposals
2.5 Chapter summary

3. Empirical Evidence Concerning the Usefulness of the Normative Proposals for Accounting Systems 10300

3.1 Market Efficiency
3.2 Empirical evidence concerning the usefulness of CCA and CPP accounting: market reaction studies
3.3 Empirical evidence concerning the usefulness of CCA and CPP accounting: studies of preparer, investor and analyst opinions
3.4 Chapter summary

Appendix 3.1 Problems with two-stage regression
Appendix 3.2 Calculation of F-values from $R^2$ statistics


4.1 An overview of computer-based accounting systems in practice
4.2 A detailed review of five computer-based TPAS
4.3 Chapter summary

5. Computer-based Accounting: Theory 4100

5.1 Textbooks on computer-based accounting
5.2 Journal articles on computer-based accounting
5.3 McCarthy's REA accounting model
5.4 Chapter summary

Appendix 5.1 A database designer's view of a GL accounting system
PART B: AN ARCHITECTURE FOR COMPUTER-BASED ACCOUNTING

   6.1 Design Principle 1: Inflation/current value accounting
   6.2 Design Principle 2: The corporate database
   6.3 Reconciling the above two Design Principles with the Objectives of Accounting identified in Chapter 2
   6.4 An Architecture for Computer-based AIS
   6.5 Chapter summary
   Appendix 6.1: Reconciling the Design Principles with the Objectives of Accounting from Chapter 2

PART C: DETAILS OF THE PROPOSED ARCHITECTURE

7. Formula Accounting
   7.1 Examples of HC/GAAP Formula Accounting
   7.2 Examples of Inflation/Current Value Accounting
   7.3 Valuing Fixed-Interest Liabilities at Market Yields, and the Use of Spreadsheets to Represent Formulae
   7.4 Comparing Formula Accounting and Constant-Journal-Entry Accounting
   7.5 Implementation of Formula Accounting in today's General Ledger software, i.e., in a system without an Interpreter
   7.6 Chapter summary
   Appendix 7.1: Capturing the Adjustments called Adj1 and Adj2 in the Spreadsheet in Figure 7.14

8. A Prototype Formula Accounting System
   8.1 A NIAM Data Model for Formula Accounting
   8.2 A Relational Model for Formula Accounting
   8.3 Implementing the Relational Database in Prolog
   8.4 Representing Formulae in the Prototype Formula Accounting System
   8.5 The Prototype Formula Accounting System
   8.6 Chapter summary
   Exhibit 8.1: An Example of the Prototype Formula Accounting System in Action

9. The Resources and Exchange Events Accounting Model
   9.1 The Resources and Exchange Events Model: An Overview
   9.2 The Exchange Events part of the REE model
   9.3 Resources in the REE model
   9.4 Comparison of the REE and REA accounting models
   9.5 Chapter summary
   Appendix 9.1: Definitions and Commentary on Terminology used in Chapter 9
10. **The Interpreter - Encoding Accounting Rules**

10.1 A Simple Historical Cost Interpreter: Interpreter 1
10.2 Enhancing Interpreter 1
10.3 The Prototype Interpreter
10.4 The Full Database used by the Interpreter
10.5 Chapter Summary and Evaluation of the Proposed Architecture

**Exhibit 10.1:** Four Runs of the Interpreter: HC and CC/CPP Output for the Exchange Events in Tables 9.1 and 9.2 (50 pages)

**Appendix 10.1:** Definitions and Commentary on Terminology used in Chapter 10

11. **Summary and Conclusions**

11.1 Detailed chapter summary
11.2 This thesis's contribution to knowledge
11.3 Suggestions for future research

**Total words:** 83,800.00

**Appendices (all Prolog code)**

1 Prolog Code for Formula Accounting 19 pages
2 Prolog Code for the Core Interpreter 28 pages
3 Prolog Code for the Special Event Handlers 10 pages
4 Low-level Prolog Code used by the above 8 pages
5 Prolog Tables for the Trading Database 20 pages
6 Prolog Tables for the Manufacturing Database 20 pages
Chapter Structure Diagram
for
AN ARCHITECTURE FOR COMPUTER-BASED ACCOUNTING INFORMATION SYSTEMS
Standard Abbreviations

(based on Tweedie and Whittington [1984], p. xi)

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>American Accounting Association</td>
</tr>
<tr>
<td>AAANZ</td>
<td>Accounting Association of Australia and New Zealand</td>
</tr>
<tr>
<td>AARF</td>
<td>Australian Accounting Research Foundation</td>
</tr>
<tr>
<td>AAS</td>
<td>Australian Accounting Standard</td>
</tr>
<tr>
<td>ACM</td>
<td>Association for Computing Machinery (US computer society)</td>
</tr>
<tr>
<td>ACS</td>
<td>Australian Computer Society</td>
</tr>
<tr>
<td>AICPA</td>
<td>American Institute of Certified Public Accountants</td>
</tr>
<tr>
<td>AIS</td>
<td>Accounting Information System(s)</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
</tr>
<tr>
<td>APB</td>
<td>Accounting Principles Board (the professional standard-setting body in the U.S. before the FASB was created)</td>
</tr>
<tr>
<td>ARS</td>
<td>Accounting Research Study (by the Research Committee of the AICPA)</td>
</tr>
<tr>
<td>ASA</td>
<td>Australian Society of Accountants</td>
</tr>
<tr>
<td>ASC</td>
<td>Accounting Standards Committee (the U.K. professional standard-setting body)</td>
</tr>
<tr>
<td>ASOBAT</td>
<td>A Statement of Basic Accounting Theory [AAA, 1966]</td>
</tr>
<tr>
<td>ASR</td>
<td>Accounting Series Release (issued by the SEC)</td>
</tr>
<tr>
<td>ASRB</td>
<td>Accounting Standards Review Board (the body which gives force of law to accounting standards in Australia; merged, in September 1988, with the AARF)</td>
</tr>
<tr>
<td>BCS</td>
<td>British Computer Society</td>
</tr>
<tr>
<td>CAPM</td>
<td>Capital assets pricing model</td>
</tr>
<tr>
<td>CBAIS</td>
<td>Computer-based Accounting Information Systems</td>
</tr>
<tr>
<td>CCA</td>
<td>Current Cost Accounting</td>
</tr>
<tr>
<td>CCE</td>
<td>Cash and Cash Equivalent (similar to CoCoA, advocated by Lee)</td>
</tr>
<tr>
<td>CC/CPP</td>
<td>Current Cost/Constant Purchasing Power</td>
</tr>
<tr>
<td>CCF/CO</td>
<td>Current Cost income from continuing operations</td>
</tr>
<tr>
<td>CoCoA</td>
<td>Continuously Contemporary Accounting (advocated by Chambers)</td>
</tr>
<tr>
<td>CODASYL</td>
<td>Committee on Data Systems Languages (a committee of the ACM)</td>
</tr>
<tr>
<td>CPP</td>
<td>Constant Purchasing Power (sometimes used in the literature to mean Current Purchasing Power, which is CPP in today's terms)</td>
</tr>
<tr>
<td>ED</td>
<td>Exposure Draft (of an accounting standard)</td>
</tr>
<tr>
<td>EMH</td>
<td>Efficient markets hypothesis</td>
</tr>
<tr>
<td>FA</td>
<td>Formula Accounting</td>
</tr>
<tr>
<td>FAGL</td>
<td>Formula Accounting general ledger</td>
</tr>
<tr>
<td>FAJE</td>
<td>Formula Accounting journal entry</td>
</tr>
<tr>
<td>FASB</td>
<td>Financial Accounting Standards Board (the private sector body which sets accounting standards in the U.S.)</td>
</tr>
<tr>
<td>FIFO</td>
<td>First in, first out (inventory valuation)</td>
</tr>
<tr>
<td>HC</td>
<td>Historical Cost</td>
</tr>
<tr>
<td>HCIFCO</td>
<td>Historic Cost income from continuing operations</td>
</tr>
<tr>
<td>GAAP</td>
<td>Generally Accepted Accounting Principles (the currently accepted set of rules for financial reporting in the US, defined today by the FASB and SEC)</td>
</tr>
<tr>
<td>ICAA</td>
<td>Institute of Chartered Accountants in Australia</td>
</tr>
<tr>
<td>ICAEW</td>
<td>Institute of Chartered Accountants in England and Wales</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>IFCO</td>
<td>Income from continuing operations</td>
</tr>
<tr>
<td>IFIP</td>
<td>International Federation for Information Processing (members include ACM, BCS and ACS)</td>
</tr>
<tr>
<td>IS</td>
<td>Information Systems</td>
</tr>
<tr>
<td>ISO</td>
<td>International Standards Organization</td>
</tr>
<tr>
<td>LIFO</td>
<td>Last in, first out (inventory valuation)</td>
</tr>
<tr>
<td>MIS</td>
<td>Management information system</td>
</tr>
<tr>
<td>NAA</td>
<td>National Association of Accountants, 919 Third Ave., New York</td>
</tr>
<tr>
<td>NRV</td>
<td>Net Realizable Value</td>
</tr>
<tr>
<td>NYSE</td>
<td>New York Stock Exchange</td>
</tr>
<tr>
<td>OV</td>
<td>Opportunity Value (a synonym for deprival value)</td>
</tr>
<tr>
<td>PV</td>
<td>Present Value (present value of expected future cash flows)</td>
</tr>
<tr>
<td>RHC</td>
<td>Real Historical Cost (HC restated for general changes in purchasing power)</td>
</tr>
<tr>
<td>RC</td>
<td>Replacement Cost</td>
</tr>
<tr>
<td>REE</td>
<td>Resources and Exchange Events accounting model (Chapter 8)</td>
</tr>
<tr>
<td>RRA</td>
<td>Reserve Recognition Accounting (as defined by the SEC in the US)</td>
</tr>
<tr>
<td>SEC</td>
<td>Securities Exchange Commission (the government standard-setting body in the US)</td>
</tr>
<tr>
<td>SFAC</td>
<td>Statement of Financial Accounting Concepts (issued by the FASB)</td>
</tr>
<tr>
<td>SFAS</td>
<td>Statement of Financial Accounting Standards (issued by the FASB)</td>
</tr>
<tr>
<td>SSAP</td>
<td>Statement of Standard Accounting Practice (issued by the ASC in the UK)</td>
</tr>
<tr>
<td>UK</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>US</td>
<td>United States of America</td>
</tr>
</tbody>
</table>
Introduction
AN ARCHITECTURE FOR COMPUTER-BASED ACCOUNTING INFORMATION SYSTEMS

Chapter 1: Introduction

Watts and Zimmerman use the term "positive accounting theory" to refer to the thrust of most empirical accounting research conducted in the last two decades. They say [1990, p.148]:

We adopted the label "positive" from economics where it was used to distinguish research aimed at explanation and prediction from research whose objective was prescription.

However, the term "positive accounting" is frequently used more narrowly to refer to work based on agency/contracting theory [Jensen and Meckling, 1976] and the "economic consequences" of accounting rule choices [Watts and Zimmerman, 1978; Whittred and Zimmer, 1990]. In his recent criticism of accounting courses Kinney [1990] indicates his strong support for this more narrow form of "positive accounting". He says that his own accounting course, completed in the US in the mid-1960s, was deficient because it over-emphasized measurement and ignored the predominantly social nature of accounting:

... my accounting courses did little to help me understand the social nature of real-world accounting and attestation or the context in which they arise. What was missing, I believe, is recognition that accounting and attestation are social phenomena. Accounting arises because of people. It affects people and is affected by people. It involves more than a bookkeeper and a set of inanimate objects to be measured. There
are also preparers, users, attestors, regulators and other outsiders, and there are contracts, conflicts of interest and markets between them and among them.

There is little doubt that the major problems in accounting are more to do with conflicts of human interest and differences of opinion than with rules for asset valuation and accounting measurement. As G.O. May, Senior US Partner of Price, Waterhouse & Co. from 1911 to 1936, said almost 50 years ago:

As far as accounting information is concerned, it seems to me fundamentally important to recognize that the accounts of a modern business are not entirely statements of fact, but are, to a large extent, expressions of opinion based partly on accounting conventions, partly on assumptions, explicit or implicit, and partly on judgement. [May, 1936, Volume 2, p.52].

May was discussing the US Securities Bill, 1933 (which together with the Securities Exchange Act, 1934, led to the formation of the US Securities Exchange Commission), but his comments are still valid today. Accounting information is prepared by some, for use by others, and conventions like the use of historical cost accounting, and the demand for auditing, have evolved to reduce the scope for "creative accounting" [Griffiths, 1986].

But whilst recognizing that accounting is first and foremost a social activity, an activity driven by human wants and desires that cannot possibly be encoded into a computer system, the focus of this thesis is nonetheless on accounting record-keeping techniques, on problems of measurement, and the processes of "doing" accounting. If Watts and Zimmerman are "positivists", the philosophical stance of this thesis might best be described as post-positivist. Post-positivists in
accounting research recognize the importance of empirical studies that compare GAAP-generated numbers to alternative measures of corporate performance (e.g., abnormal market returns and corporate bond ratings), or that explain choices in accounting techniques in terms of contracting theory and its economic consequences. But post-positivists also recognize that there are limitations to what these studies can achieve. It is useful to know that GAAP earnings both reflect stock market information and convey information to the markets [Ball and Brown 1968; Foster 1977], and it is important to know that the current cost accounting (CCA) reports of SFAS33 [FASB 1979] conveyed very little information to US stock markets beyond that conveyed by GAAP accounts [Beaver and Landsman 1983; Bublitz, Frecka, and McKeown 1985], but there is still room in accounting for research into better ways to measure and report economic performance. In their attempts to bring a new order to accounting, post-positivists draw upon both empirical studies that predict and explain what is, and normative studies that argue for what should be.

1.1 Statement of Objectives

The Problem addressed in this thesis is whether cost-effective, computer-based accounting systems can be used to generate better accounting information than existing transaction processing accounting systems. It is clear from the widespread use of computers in accounting (see evidence in Chapter 4) that computers have "arrived" in accounting practice. What is not clear is how computers should be used. Designers of transaction processing systems in the past have tended to focus on producing information more cheaply, or faster, but
few questions have been asked about whether better accounting information can be generated using a computer.

Asking questions about better accounting information leads to questions that a normal systems designer would not have the time or background to consider: questions like how assets should be valued for different decisions, whether managers can be trusted as sources of values for accounting statements, and whether it is worth collecting net realizable values for accounting reports. One of the questions asked in this thesis is whether double-entry bookkeeping, a technique that developed in the northern cities of Italy in the fifteenth century, is still the best conceptual model for accounting today. It could be that double-entry bookkeeping leads to certain ways of thinking that are unproductive. For instance, it might tend to perpetuate the use of historical cost accounting, which, in times of inflation may be undesirable. A normal systems analyst with deadlines to meet would just assume the status quo, and focus on operational issues like screen layout, security during on-line update, audit trails, reporting flexibility, and so on. This thesis asks more fundamental questions. In Chomsky's [1965, 1979] terminology, its concern is with the deep structure of accounting systems, not their surface structure.

The reason it is worth asking "deep structure" questions, like those mentioned above, is that computer technology has changed the way most Western organizations store and process information. Now that accounting records are stored in machine-readable form it is possible to manipulate information, and to generate reports, in ways that were not possible using the hand-written records of the fifteenth century. Yet although almost all large firms maintain their
accounting records using computer-based systems, the underlying conceptual framework of ledger accounts and debits and credits, would still be recognizable to Pacioli [1496]. It may be that ledgers and journals are still the best way to conceptualize accounting, but having witnessed the changes in other areas accompanying the widespread introduction of computing, it seems reasonable to ask whether this very powerful force for change, computing, might also lead to some fundamental changes in the way accounting is done. Thus this thesis approaches the design of computer-based transaction processing accounting systems from first principles. It attempts to leave no stone unturned in its efforts to understand why accounting systems exist, what accounting systems should do, and whether some alternative method of accounting might do the job better.

1.2 Structure of this thesis

The structure of this thesis is unconventional because it is not concerned with theory building and testing. According to Kerlinger [1986, Appendix D] a conventional theory-testing research report should be structured as follows:

1. The Problem
2. Methodology and Data Collection
3. Results, Interpretation, and Conclusions.

In this conventional structure, The Problem section reviews the literature and states the hypotheses to be tested. The Methodology and Data Collection section explains the research design to be used for hypothesis testing, and gives
the reader sufficient details of data collection for the study to be replicated. The Results, Interpretation, and Conclusion section reports on the results of hypothesis testing and identifies major limitations of the study.

However, this thesis is concerned with design, and a conventional theory-testing structure is inappropriate. Design is a creative activity that Simon [1960, p.54] described as the process of "inventing, developing, and analyzing possible courses of action". The designer first identifies a problem, then sifts through the myriad possible solutions looking for combinations that best solve the problem. Like a theory, a design may be tested -- the Wright brother's design for an aeroplane could be tested -- but most of the effort in design goes into the defining, searching, matching, creating stage, not the testing. Again in Simon's words, design involves processes "to understand the problem, to generate solutions, and to test solutions for feasibility" [Simon 1960, p.54].

Thus, instead of using the theory-testing structure above, this thesis is structured as shown in the Chapter Structure Diagram in Figure 1.1 (next page). In essence, it consists of three main Parts, A, B, and C, as follows:

1. Introduction (this chapter)
2. Part A: Requirements Determination (Chapters 2 - 5)
3. Part B: Synthesis: The Proposed Architecture (Chapter 6)
4. Part C: Details of the Proposed Architecture (Chapters 7 - 10)
5. Summary and Conclusions (Chapter 11)

1 - 6
The objective of Part A is to determine the requirements for future computer-based accounting information systems. Part B is a single chapter that reviews the evidence collected in Part A, and from that, develops the proposed architecture for computer-based accounting. Part C consists of four chapters that probe the feasibility of the proposed architecture. Fuller details of each Part are given in Sub-Sections 1.2.1 through 1.2.3 below.
Although this thesis structure is unconventional, there is still a Problem to be solved, there are still data to be collected, and there is still a conclusion or resolution of The Problem to be presented. However, the contribution to knowledge is different. In a theory-testing thesis, the contribution to knowledge comes in the form of a theory, supported by evidence, about how the world works. The contribution may be the theory, the method of collecting and testing evidence (a four-step process of hypothesis formulation, research design, data collection, and statistical analysis), or both. In a design thesis, the researcher contributes to knowledge, first, by providing clearer insights into deficiencies in current ways of doing things, and second, by suggesting novel but practical solutions to those deficiencies. In both cases the test of a thesis's "original contribution to knowledge" is evaluated by asking the same questions:

**Originality:** In what ways are the ideas in this thesis different to those already reported in the literature?

**Contribution:** How important are these new ideas to the discipline?

Details of the original contributions claimed for this thesis are given in Chapter 11, Section 11.2.
1.2.1 Part A: Requirements Determination

Although Part A is headed "Requirements Determination", it is not conventional Requirements Determination where, according to Davis and Olson [1985, p.441], "Users are the fundamental source of requirements". In this design project, there are no users, there are no issues like the socio-technical "fit" of the system in the organization to be considered [Mumford, 1981; Wood-Harper, Antill, and Avison, 1985], and there are no questions to be asked about the relative power of various players in the organization [Markus, 1983]. Instead, requirements are determined through was is essentially a mammoth literature survey.

This literature survey is organized as two pairs of two chapters exploring two independent themes. Chapters 2 and 3 focus on the question of what constitutes the "better" accounting information that new computer-based accounting systems might be asked to produce. Chapter 2 examines the normative accounting literature. It considers the objectives of accounting, problems with accounting measurement, and summarizes the major proposals for accounting reform that dominated the accounting literature in the 1960s. Chapter 3 explores the positive accounting literature for evidence to test the merit of the normative suggestions from Chapter 2. These two chapters are based firmly in the mainstream accounting literature, and are probably responsible for two thirds of the references in the bibliography.

Chapters 4 and 5 attempt to define the current state of computer-based accounting; this might be likened to the "documentation of the existing system"
that is often part of a conventional systems design exercise. Chapter 4 examines computer-based accounting as it is practiced today. It contains a summary of findings from two questionnaires mailed to Australian firms during 1988 and 1989. It also describes some quite sophisticated present-day accounting software in some detail. Chapter 5 summarizes the (sparse) academic literature on computer-based accounting.

1.2.2 Part B: Synthesis - An Architecture for Computer-based Accounting

Part B of the thesis is a single chapter where the two streams of thought from Part A are brought together, some hard decisions are made, and an architecture for computer-based accounting is proposed.

Sections 6.1 and 6.2 assess the implications of the preceding four chapters in an attempt to define requirements for the next generation of computer-based accounting information systems. These requirements, which form the starting point for Section 6.4, are presented in the form of the following two Design Principles:

**Design Principle 1: Inflation Tolerance**

It is worth trying to build a cost-competitive, general-purpose, computer-based accounting system that is able to generate reports in inflation-adjusted/current-value terms.
Design Principle 2: The Corporate Database

Since data must be collected for the firm's special-purpose feeder systems for day-to-day control purposes, they should be retained in machine-readable form as a corporate database for subsequent analyses.

These Design Principles are both normative recommendations, i.e., they are just assertions about how accounting should be done. They are impossible to prove or disprove. The arguments presented in Sections 6.1 and 6.2 are supposed to be logical, but someone who perceived the objectives of accounting differently might well come to different conclusions.

In Design Principle 1 the term "inflation" refers to any form of price change accounting, not just general inflation. Strictly, "inflation" should only be used to refer to a movement in the general level of prices. However, in the literature today, e.g., Baxter [1984] and Tweedie and Whittington [1984], the term "inflation accounting" is often used as a generic term, covering any sort of price-change accounting. This terminology is now in such wide use that in this thesis such terms as "inflation accounting", "inflation-adjusted" accounting, and "inflation-tolerant" accounting are used as generic terms referring to any form of the price-change/current-value accounting discussed in Chapter 2, i.e., CPP, CCA, CoCoA, and CC/CPP.

Design Principle 2 arises from the observation that many different, special-purpose feeder systems are required for different organizations, but that the needs of these systems are so diverse and ever-changing that it is impractical to
suggest that they should be standardized. It is therefore suggested that the most fruitful place to look for generalized improvements to accounting systems is in the area of the general ledger.

Figure 1.2: The Proposed Architecture for Computer-based Accounting

Section 6.4 (still in Part B of the thesis) accepts the above two Design Principles as given, and asks how data in the corporate database (Design Principle 2) might best be used to achieve inflation-tolerant accounting reports (Design Principle 1). The problem, and its proposed solution, are depicted in Figure 1.2. Figure 1.2 is the proposed architecture for computer-based accounting that motivated the title for this thesis. On the left are the databases from the various subsystems. On the right are "inflation-adjusted accounting reports". In the middle are two suggested mechanisms for using data from the subsystem databases, combined with "special information" such as replacement costs and price index series, for generating inflation-adjusted accounting reports.
1.2.3 Part C: Details of the Proposed Architecture

Part C of the thesis consists of four chapters, Chapters 7 through 10, that develop the skeletal proposals in Figure 1.2 into a practical system of accounting. Most of the material in Part C is new, so there are far fewer references to the literature than in Parts A and B of the thesis.

Chapters 7 and 8 show how an improved technique for double-entry accounting, called Formula Accounting, could be used to enable general ledger systems to model changes in value much more simply than in present-day systems. Formula Accounting is one of the major original contributions of this thesis.

Chapters 9 and 10 show how journal entries for a Formula Accounting general ledger could be generated using a new data model (called the Resources and Exchange Events accounting model) and a program (called an Interpreter) that contains the rules necessary for generation of the journal entries. An overall system consisting of a Formula Accounting general ledger fed by an Interpreter using data from an REE model is called a Multiview Accounting system because once the REE model has been implemented one or more Interpreters could be used to generate multiple accounting views of the one set of economic events.

It is difficult for a human being to examine a complex design like that proposed in Chapters 7 through 10 and have much sense of the limitations of the system. Much is promised, but will it work? To show that the proposed architecture does work, that the REE model is sufficiently complete for Formula Accounting journal entries to be generated, and to reduce problems of misunderstanding
that inevitably arise when abstract ideas are discussed, a prototype Multiview Accounting system was developed. Chapters 8 and 10 give details of that prototype, which was implemented in a programming language called Prolog. The working prototype demonstrates the feasibility of the proposed architecture and provides evidence that the underlying data model is reasonable complete.

For completeness, a full listing of the Prolog code for the prototype system is included in a series of appendices to the thesis, Appendices 1 through 6. It is not intended that the reader should read the code in any detail, but as output from that code is included in significant appendices to Chapters 8 and 10, the code must clearly be available to show how the output was generated. A diskette containing the same code is also included with the thesis. The code is written in standard Clocksin and Mellish [1981] Prolog, and should be executable by most Prolog interpreters with minimal modification.

1.2.4 Chapter 11: The Thesis Summary

As noted earlier, the topic addressed in this thesis is much broader than is usually attempted in a PhD thesis. So many issues are covered that it is difficult to keep track of them all at one time. Each chapter therefore has a Chapter Summary section, and the Chapter Structure Diagram presented earlier, Figure 1.1, shows how the chapters fit together.

Because the issues covered are so wide-ranging, Section 11.1 in the final chapter also contains a quite detailed summary of the arguments presented in the body
of the thesis. Section 11.2 documents the main contributions to knowledge, and Section 11.3 contains some suggestions for future research.

(3,100 words)
Part A: Requirements Determination
The purpose of this chapter is to identify those characteristics of accounting information that are thought to make one system of accounting better than another. Since the material is based on eighty years of accounting research, by countless researchers, it is clearly neither possible nor desirable to review all the arguments, or even the main arguments, used by advocates of all the different competing accounting systems. For instance, it would be difficult to improve on Henderson and Peirson's [1983] 268-page thorough, balanced, well-documented portrayal of the evolutionary development of accounting thought from the "pre-theory period" (1494-1800), through the "general scientific theory period" (1800-1955), and the "general normative theory period" (1956-1970), to the "scientific theory period" (post-1970). The approach adopted for this chapter is therefore to aim for a concise summary of the products of normative accounting theory. The objective is to search for common ground wherever possible, without ignoring important differences. What is reported is intended to cover most of what is said to be "good", "useful", or "desirable" about accounting information.

The four-step structure of this chapter is depicted in Figure 2.1 (next page). To begin with, a search of the literature identifies three main Objectives of Accounting. This list of objectives, which is presented and discussed in Section 2.1, is intended to be simple yet reasonably complete. These objectives define the domain of discourse for this thesis. In Step 2, Section 2.2, the information needed to meet these objectives is discussed. In Step 3, Section 2.3, the data
March 3, 1991

Chapter 2: Normative Accounting Theory

The purpose of this chapter is to identify those characteristics of accounting information that are thought to make one system of accounting better than another. Since the material is based on eighty years of accounting research, by countless researchers, it is clearly neither possible nor desirable to review all the arguments, or even the main arguments, used by advocates of all the different competing accounting systems. For instance, it would be difficult to improve on Henderson and Peirson's [1983] 268-page thorough, balanced, well-documented portrayal of the evolutionary development of accounting thought from the "pre-theory period" (1494-1800), through the "general scientific theory period" (1800-1955), and the "general normative theory period" (1956-1970), to the "scientific theory period" (post-1970). The approach adopted for this chapter is therefore to aim for a concise summary of the products of normative accounting theory. The objective is to search for common ground wherever possible, without ignoring important differences. What is reported is intended to cover most of what is said to be "good", "useful", or "desirable" about accounting information.

The four-step structure of this chapter is depicted in Figure 2.1 (next page). To begin with, a search of the literature identifies three main Objectives of Accounting. This list of objectives, which is presented and discussed in Section 2.1, is intended to be simple yet reasonably complete. These objectives define the domain of discourse for this thesis. In Step 2, Section 2.2, the information needed to meet these objectives is discussed. In Step 3, Section 2.3, the data
needed to provide that information are considered. It turns out that the data needed for most recommended systems of accounting are far more homogeneous than might, at first, be expected. For instance, whilst management estimates of future cash flows are used routinely for internal capital investment decisions, few people seriously suggest that these figures should be used for external reporting of economic performance - there is too much room for error and manipulation. Finally, in Step 4, Section 2.4, five major systems of accounting are identified. Each is tied back to the data that must be collected to enable the system to work. This is important because data are not costless nor equally reliable. For example, CoCoA [Chambers 1966] requires that net realizable values of assets be estimated for each accounting report. Both the cost and reliability of CoCoA data need to be considered when deciding whether CoCoA is a "better" system than, say, historical-cost accounting.

<table>
<thead>
<tr>
<th>2.1</th>
<th>Objectives of Accounting</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Information needed to meet those Objectives</td>
</tr>
<tr>
<td>2.3</td>
<td>Data needed to provide that information</td>
</tr>
<tr>
<td>2.4</td>
<td>Five major systems of accounting that use those data to achieve the Objectives</td>
</tr>
<tr>
<td>2.5</td>
<td>Chapter summary</td>
</tr>
</tbody>
</table>

Figure 2.1 Section-chaining structure of this chapter
2.1 The Objectives of Accounting

Most texts on Accounting Theory contain a section on the objectives of accounting. Key contributions in the last thirty years have been made by Moonitz [1961], Sprouse and Moonitz [1962], the ASOBAT committee [AAA, 1966], the AICPA [1970, APB 4], and the FASB [SFAC1, 1978 and SFAC2, 1980a]. From this vast literature it is possible to distil three main objectives for accounting. These are presented in Figure 2.2 and discussed briefly below.

1. To provide management with information useful for planning and control.

2. To provide external users with information useful for predicting the future cash flows and the earning power of a firm.

3. To provide accountability information useful for contract monitoring.

Figure 2.2: The Objectives of Accounting

Before commencing a detailed discussion of these Objectives and their data requirements it is necessary to indicate the presumptions that underlie them. First, there is almost universal agreement, from both academics and practitioners, that the all-encompassing, main objective of accounting is to provide information useful for economic decisions. Edwards and Bell [1961, p.25], Staubus [1961, p.11; 1977, p.20-31], the ASOBAT committee [AAA, 1966, p.1], the AICPA (APB Statement No. 4, [1970, para 40]; Trueblood Committee
have all taken this as their starting point for discussion. All three objectives in Figure 2.2 are consistent with this main objective.

Second, it is widely agreed that accounting reports are not the only sources of information available to economic decision-makers, inside or outside a firm. Ball and Brown [1968], the FASB [SFAC 1, 1978], Foster and Horngren [1988, p.5], and Beaver [1989, p.12] all make this point explicitly. The main uses of accounting information must therefore be for comparing actual performance against predictions, and as authoritative sources of data for further predictions. For example, according to Brown, Foster, and Noreen [1985], stock market analysts are able to make quite accurate estimates of accounting income in the months before the actual announcements, but it is the actual number from the accounts that is always regarded as the authoritative figure.

The observation that accounting reports are not the only source of information for economic decision-making suggests limits to what accountants should try to measure. Human resource accounting (HRA) might be "off-limits", for example. The objectives of HRA might be better met by other means. By this test, all three objectives in Figure 2.2 are well within the limits of the information that accounting is normally supposed, and is able, to provide.

Third, it is widely recognized that accounting information is used by many different people for many different sorts of economic decisions. The prime distinction is between information for (a) management and (b) external reporting.
2.1.1 Information for management

It will be argued in Chapter 4, that present-day computer-based accounting systems are designed primarily for internal reporting to management\(^1\). As Edwards and Bell say,

"the bulk of accounting data is never made available to people outside of the business firm itself", so "it seems safe to conclude that accounting information must principally serve the functions of management" [1961, p.4].

Objective 1 in Figure 2.2 is clearly consistent with this management accounting emphasis. However, senior management of large firms take a keen interest in the reactions of external users, e.g., investors and stock market analysts, to their periodic reports\(^2\). Owner-managers of small firms are also, of course, interested in measuring the performance of their own firms. Thus, although computer-based accounting systems may be designed primarily with Objective 1 in mind, the needs of financial accounting and statutory reporting will also influence the way they are designed. Financial accounting information is derived from the same computer-based accounting system that is used for management accounting, so problems of financial accounting, i.e., asset valuation, income measurement, and inflation, are also problems that designers of computer-based accounting systems must address.

---

\(^1\) Cushing's suggestion [1989] that future accounting systems may allow public access to a firm's accounting database is a possibility, but it is unlikely that management would allow access to more than highly summarized data.

\(^2\) Critics of "creative accounting" [Griffiths, 1986] say that some managers are too interested.
2.1.2 Information for external users

There seem to be two major objectives for external financial reporting. One is to assist investors and creditors predict a firm's future earnings and cash flows. This includes providing feedback information about the success of prior predictions. The second is for accountability reporting. Ijiri [1975, p.32] describes accountability, an extension of the stewardship concept, as having been "the social and political backbone of accounting for centuries". This dual role for financial accounting is reported in many studies. The following four studies indicate the general tenor of discussion:

(i) The US Trueblood Committee [AICPA, 1973] prepared a list of twelve objectives of financial statements. First came the all-encompassing objective presented above, i.e., the provision of "information useful for making economic decisions". Next followed eleven sub-objectives for financial statements. Six of the eleven suggest that many users of financial reports are interested in "predicting, comparing, and evaluating" the cash flows and "earning power" of various firms. The remaining five sub-objectives are difficult to classify, but Sorter, the Research Director for the Trueblood Committee, classifies Objective 5 as an "accountability" objective [Sorter and Gans, 1974]. In other words, the Trueblood Committee recognized both Objectives 2 and 3, but gave more weight to the predictive than the accountability objective.

(ii) One year after Trueblood, Carsberg, Hope and Scapens [1974] in the UK listed the same two objectives (in reverse order of importance):
The traditional stewardship objective is still widely acknowledged as important. There appears to be a growing consensus, however, that the provision of information to assist shareholders with their investment decisions should be recognized as a second important objective of accounting statements.

(iii) More recently, the FASB, in Statement of Financial Accounting Concepts No. 1 [1978] has again repeated the same two reasons for financial accounting. In the words of Wolk, Francis and Tearney [1989, p.177], SFAC 1 first acknowledges the "heterogeneity of external user groups", but decides that

... a common core characteristic of all outside users is their interest in the prediction of the amounts, timing, and uncertainties of future cash flows. ... The statement also notes the importance of stewardship in terms of assessing how well management has discharged its duties and obligations to owners and other interested groups.

(iv) Whittred and Zimmer [1990, p.11] take exception to the Trueblood, Carsberg et al., and SFAC 1 position. Their agency-theory-based view of accounting treats financial information and reporting as

... being contracted for in advance as part of the mechanism by which the agent restricts the extent to which his or her actions deviate from those required by the principals.

They note that "the agency view and the decision usefulness view are not mutually exclusive" (p.12), but they clearly believe that the agency approach leads to simpler explanations of the choices people make between present or proposed alternative accounting rules.
Clearly there is some controversy about the relative importance of the decision-usefulness and contracting roles of financial accounting, but equally clearly both Objectives 2 and 3 in Figure 2.2 are fundamental objectives of financial reporting.

2.2 Information Required to Meet the Three Objectives from Section 2.1

2.2.1 Objective 1: To provide management with information useful for planning and control.

A striking feature of the computer-based accounting systems described in Chapter 4 is the emphasis on detail: detailed budgeting, and detailed historical records for comparison with current performance. For instance, the largest general ledger system examined, that of Australia’s largest retailer, Coles Myer Ltd, had 700,000 accounts. Each of these accounts contained four years of actual and budget monthly summary figures, i.e., over 50 million pieces of data. Managers apparently use this, and more, accounting information for target setting, capital budgeting, pricing decisions, and for cost control. Apart from this desire for detailed information, it is difficult to anticipate the information needs of management. Their needs are so diverse. It follows that for systems designed to meet Objective 1, the best approach is probably to establish a database of detailed, potentially relevant, facts that is available when needed. This invites questions about how one decides which facts are potentially relevant and worth collecting.
The choice of facts to be stored is partly determined by the desire to meet the Accountability objective, Objective 3. It is in pursuit of the Accountability objective that accounting systems today record detailed transaction histories of all financially relevant events. Once details of all transactions like purchases and sales (of inventory and fixed assets), wages payments, expenses of various kinds, capital raising, and details of goods input to, and output from, various stages of a manufacturing process, have been recorded, the needs of management for planning and control are likely to be fairly firm-specific. An information system designed to optimize inventory holding costs in many warehouses across a country may be important to management planning and control for the firm that uses it, and have important income effects for that firm, but specific details of the system are not generally useful to all firms. In addition, of course, much of the data used for non-routine planning and control will still have to be collected as and when required, often from outside the firm.

Implications of Objective 1 for AIS Design:

These considerations lead to two important conclusions about the importance of management planning and control information that limit the scope of this thesis as shown in Figure 2.3 (next page). First, there is no doubt that special-purpose systems are an important part, if not the most important part, of management accounting/ MIS for most firms. But since they are firm-specific, or rely on ad hoc data, it is simply not possible to make any useful generalizations about them. This thesis will therefore concentrate only on routine information for management planning and control. This is the middle box in Figure 2.3. Second, it will be suggested in Chapter 6 that as computer-based accounting systems
evolve, most information for routine day-to-day planning and control will tend to come from sub-systems chosen or tailored to meet the specific information needs of different firms. Whilst it should be possible to use the databases of these firm-specific sub-systems as rich sources of data for both ad hoc planning and conventional accounting reporting, it is not possible to make any strong generalizations about the design of such systems. It must therefore be concluded that the most fruitful place for widely-generalizable research into accounting information for management planning and control is at the level of the general ledger. This is the inner-most box in Figure 2.3.

**Figure 2.3 Restricting the Scope of 'Objective 1'-Information for this Thesis**

Note that the decision to focus on the general ledger information for management planning and control does not ignore the importance of the firm's special-purpose sub-systems. These sub-systems are likely to be the prime source of data for management accounting information. Summary data from these systems can be funnelled into the firm's general ledger.
2.2.2 Objective 2: To provide external users with information useful for predicting the future cash flows and the earning power of a firm.

While computer systems can store detailed transaction histories for many years, and classify and summarize that information relatively easily, this storage and processing power provides no easy answers to the problem posed by Objective 2. Satisfying Objective 2 has been, is, and probably always will be, the most difficult task faced by accountants. What information is relevant? Various ways of meeting Objective 2 have been suggested, some of which are discussed below, but by far the most important suggestion is that of income measurement.

Income measurement is supposed to assist both with predicting future cash flows and with measuring past performance. It would be possible to try to meet the forecasting part of Objective 2 by reporting estimated future cash flows directly. But because it is easy to over-estimate future cash flows, possibly deliberately, few people are prepared to trust an entrepreneur's or manager's cash flow estimates. Income measurement is supposed to avoid that problem by being based on past events. Unfortunately there is no direct theoretical link between any of the normal accounting income measures, like HC, GAAP, CPP, CCA, and NRV income, and future cash flows.

For the external user, Sharpe [1985] reminds readers that it is dividends, not earnings, that determine share value. "Earnings", he says, "are important, but only because they can provide dividends" [p.416]. Sharpe links stock prices to
earnings through the price earnings ratio. His equation 14-6 [p. 425] says that a stock's normal price-earnings ratio is determined as follows:

$$\frac{P_0}{e_0} = \frac{k_1(1+g_1)}{1+r} + \frac{k_2(1+g_1)(1+g_2)}{(1+r)^2} + \frac{k_3(1+g_1)(1+g_2)(1+g_3)}{(1+r)^3} + \ldots$$

where \( P_0 \) is the share price now, \( e_0 \) is the latest earnings per share, \( k_1, k_2, k_3, \ldots \) are the expected dividend payout ratios (the fraction of earnings paid as dividends) in successive future periods, \( g_1, g_2, g_3, \ldots \) are the expected earnings growth rates in successive future periods, and \( r \) is the expected appropriate discount rate (assumed constant) for the risk involved.

Multiplying both sides of Sharpe's equation by \( e_0 \) yields an equation where each term in the right-hand side is the present value of the shareholder's expected cash flow for the period. Now, in a practical context, Sharpe's equation is unrealistic: first, because it assumes it is possible to predict all the variables, and second, because it ignores the speculative element in stock prices. Yet it represents the best that theory can offer to explain a link from present earnings to a firm's future cash flows, and how investors value those cash flows.

In practice, the PV of an entire firm is sometimes estimated by (i) using the historical time series of reported HC income, together with knowledge of expected future conditions, to predict average future HC income, then (ii) capitalizing that income using a price/earnings ratio (P/E) based on those prevailing in the market place. Backer [1973, Chapter 10] provides a good example of this approach as used by an investment banker. However, it is hard to find theoretical support for the choice of a price-earnings ratio. For instance, just before the 1987 stock-market crash Arthur Andersen & Co. used a
P/E of 17 in its independent valuation of Smorgon's steel smelting business in Laverton during the takeover by Humes Ltd. Post-crash, P/E ratios were much lower. Which P/E figure is correct? Estimates of the underlying cash flows and profitability would probably not have changed markedly\(^3\) over the few days of the crash, yet the values placed on the firm are very different.

In short, the absence of any deterministic link between reported income and future dividends makes it difficult to translate Objective 2's goal of assisting investors "predict future cash flows" into rules for measuring and reporting accounting income. Notwithstanding this, most theorists still conclude that income measurement is the most useful single measure of a firm's performance that an accountant can hope to generate.

Recognizing the many different interpretations and uses of accounting income, Hendriksen [1977, Ch. 5] presents a classification of income concepts under three headings: structural, interpretive, and behavioral, as follows:

(i) According to the "structural" view, income has no meaningful economic interpretation. It should be understood only by knowing the, hopefully consistent, rules by which it is measured.

\(^3\) It is difficult to accept that investors suddenly lowered their estimates of firms' future profits, for hundreds of firms, so suddenly. More likely, when the bubble burst, hopes for quick, short-term holding gains evaporated, and speculators raced to liquidate their stock. By this argument, P/E ratios fell because prices fell, with earnings expectations remaining largely unchanged.
Under the "interpretive" view, income is a surrogate measure of increase or decrease in economic wealth\(^4\) over time, and can be used for performance evaluation. The wealth measurement approach leads to income measurement based on some concept of capital maintenance.

Finally, under Hendriksen's "behavioral" classification, income is viewed as useful for predicting cash flows, market prices of shares, solvency, and for managerial decision-making.

The first view may be an apt description of GAAP income and of "proper" matching: both are convention-driven. Capital market researchers tend to adopt this view, but argue that income is useful, nonetheless, because it can be demonstrated empirically that it both reflects market returns and conveys information to the market [Watts and Zimmerman, 1986, Ch.3]. However, a structuralist view of income is not helpful for this thesis: it does not lead to suggestions for improved measures.

The remaining two views of income are closer in spirit to meeting the requirements for Objective 2. In fact, the "interpretive" view of income, based on ideas of economists like Hicks [1946], is really not so different from the "behavioral" view. It seems unlikely that a measurement system that is useful for decision-making, in many firms, over many years, should not in some way reflect changes in wealth. Yet by the time the economist's wealth measure arrives in the accountant's balance sheet it has undergone a subtle, yet major,

\(^4\) Fisher [1906, p.52], Hicks [1946, p.172].
transformation. Witness the FASB's definition of "comprehensive income" [SFAC 3, 1980b, para.56]:

Comprehensive income is the change in equity (net assets) of an entity during a period of transactions and other events and circumstances from nonowner sources.

Whereas an economist's definition of wealth is based on expectations of future cash flows, the FASB definition of wealth implicit in its definition of comprehensive income is based on past or current costs of the subset of assets and liabilities that happens to be readily identifiable. This subtle shift in meaning may be one of the major problems with accounting systems based on wealth measurement arguments: so many components of economic wealth are omitted from accounting income measures, like GAAP, CPP, CCA, and even CoCoA income, that the resulting measures may lack the economic significance they claim.

To consider an extreme case, a mining company might discover a large ore body of such clear economic significance that its share price soars on the stock market. Neither GAAP income nor CCA income will be affected until the mine is actually producing. CCA would then focus on providing detailed information about holding gains (increases in value) of the particular classes of assets, like mining equipment, that accountants regard as acceptable to measure, but would go on ignoring the most important asset, the mineral deposit itself. As discussed in Chapter 3, the stock markets of the world have shown no preference for CCA over GAAP. Perhaps the markets' lack of interest in CCA
is because both GAAP and CCA ignore so many of the same factors important to economic-wealth measurement that CCA is of little more use than GAAP.

It is sometimes suggested that instead of using income as the measure of past performance, one might satisfy Objective 2 by reporting a time-series of past cash flows [Hicks and Hunt (eds), 1981]. Past cash flows would be of no more help in the mining company example than accounting income, but their measurement does not depend on the asset valuation rules and arbitrary allocations [Thomas, 1969, 1981] that plague income measurement. While the timing of cash flows is open to some manipulation, cash flows are nevertheless much more objective, and understandable, than any income measure. Furthermore, cash flows are an indicator of solvency, and this in itself is important. Kochanek and Norgaard [1988] describe a recent case where operating cash flows of a Fortune 100 corporation, the Charter Company, were minus $90 million in the year before the firm went bankrupt. They also show that in the same period the Charter Company was able to report income from continuing operations of $50.4 million. Admittedly this income figure is low compared to revenue of $5.7 billion (approximately 1% of revenue), but in this instance the negative net operating cash flow provided a clearer danger signal.

However, just as a time series of income numbers does not tell the whole story, neither does a series of cash flows. To make past cash flow reporting useful for predicting the future, it is often helpful to supplement the cash flow information with some sort of measure of stored wealth at the end of each period. For example, Lee [1984] recommends reporting current cash equivalents.
(Chambers, 1966, 1980) of a firm's assets. Conventional accrual accounting balance sheets provide another indicator of stored wealth.

Given the empirical support for both earnings [Ball and Brown 1968] and cash flow [Kochanek and Norgaard 1988] information, it would seem sensible, if costs were not prohibitive, to report both forms of information. The recent decision by the FASB to supplement GAAP income statements and balance sheets with a statement of cash flows [SFAS 95, 1987], in place of the previously required working-capital funds statement, therefore has much to commend it.

Implications of Objective 2 for AIS Design:

Objective 2 is probably the most difficult for an accounting system to meet. In some cases, like the mining example above, it is impossible to use past measures, income or cash flows, to predict future cash flows or earning power. In such cases investors and others simply have to turn to other information sources, e.g., management and other expert estimates of ore reserves, future demand, production levels, exchange rates, and so on, to estimate future cash flows and earning power. However, for more mature firms, measures like accounting income and past cash flows may be useful for predicting future cash flows and earning power. Moreover, accounting systems can be designed to provide both forms of information. For these more mature firms, the majority of whose shares are traded on world stock markets, five alternative income measurement systems are discussed in Section 2.4.
2.2.3 Objective 3: To provide accountability information useful for contract monitoring.

As mentioned earlier, Ijiri [1975] views the accountability objective as the paramount objective of accounting. For accountability reporting one should be able to show, at any time, exactly what the firm's assets and obligations are, how that situation came to be. Accountability must be important, because most present-day accounting systems maintain detailed records of all cash transactions, changes in inventory levels, purchases and disposal of fixed assets, and details of current and other liabilities, usually in meticulous detail. Accountability is so important, in fact, that directors of Australian companies are required to ensure that their firm maintains accounting records:

A company is required:
(a) to keep such accounting records as correctly record and explain the transactions of the company; and
(b) to keep its accounting records in such manner as will enable --
(i) the preparation from time to time of true and fair accounts of the company; and
(ii) the accounts of the company to be conveniently and properly audited in accordance with the Code.

[Companies Code (Vic.), s 267(1)]

In what follows, therefore, it will be assumed that detailed, auditable, transaction-history records are required in any future accounting system, computer-based or otherwise.

The other purpose of accounting linked with "accountability" reporting in Objective 3 is contract monitoring. Agency theorists, like Jensen and Meckling [1976], Watts and Zimmerman [1986], and Whittred and Zimmer [1988] maintain
that agents willingly enter contracts to restrict their consumption of perquisites in order to maximize their own welfare. To meet the contract-monitoring objective, accounting records need to be clear and unambiguous, with little scope for manipulation by those being monitored. Lenders, such as bond holders, who use accounting information to monitor debt contracts are usually less interested in income measures than they are in solvency. From a lender's perspective, normally conservative valuation rules like those in HC, or Chamber's CoCoA [1980], might be most the appropriate form of accounting, particularly for balance sheets.

On the other hand, agency theorists also recognize that shareholders tend to use accounting numbers in "contracts" with management. Unlike bond holders and other lenders, shareholders want to motivate their agents to maximize firm value. Ideally, shareholders would prefer bonuses and other rewards to management to be based on increases in the economic wealth, i.e., the present value of future net cash flows, of the firm. However, since economic wealth is not objectively measurable, bonus schemes must be based on other measures. Two possible bases are changes in share prices (possibly after market-wide movements have been removed), and accounting income. Most bonus plans use the latter. Which sort of income measure is most suitable for bonus compensation schemes? Most of the empirical studies reported in the next chapter, Chapter 3, found quite high correlations (of around 0.8) between CCA and HC operating income in the US and UK in the early 1980's. So it would seem that under these conditions most shareholders would find HC accounting (which is cheaper than CCA, and less open to manipulation) adequate for contract monitoring. However, in times of high inflation, HC income overstates
real margins by matching out-of-date, lower, costs against more current, higher, revenues. If shareholders expected high inflation, e.g., if they were shareholders of a Brazilian company, they might be much more inclined to prefer some form of inflation accounting for contracting purposes.

Implications of Objective 3 for AIS Design:

To meet the accountability part of Objective 3, a firm must keep a detailed, precise record of all economic resource exchanges.

The need for contract-monitoring information, also part of Objective 3, leads to conflicting requirements for accounting system design. Many people favour HC for its low cost and relative objectivity. Lenders want accountants to practice conservatism, and would probably prefer CoCoA because of its emphasis on liquidation values in the balance sheet. But shareholders, particularly in times of inflation, would probably prefer that management bonuses were paid on the basis of some form of reasonably objective inflation accounting. (GAAP accounting would tend to report higher income numbers in times of inflation.) It would therefore seem wise to keep all options open at this stage in the design.

2.3 What Data to Collect?

Because of uncertainty about the future, none of the major normative accounting systems proposals encountered during the literature survey recommends that financial accounting reports should value firms, or derive income measures, from expected cash flows from operations. Instead they
advocate an array of approaches to asset valuation, ranging from cash liquidation values through to current replacement costs. These proposals are described in some detail in the next section, Section 2.4, but before describing them, it is worth clarifying the way accountants deal with some problems common to all accounting-income-measurement systems, namely (a) the grouping problem in asset valuation, (b) measurement error, and (c) data collection for asset valuation and income determination. Asset valuation and income determination are usually linked because, all other things being equal, in an articulated accounting system increases in asset values cause increases in accounting income.

2.3.1 The grouping problem and the "Better to Disclose Details" rule

To a potential purchaser, the value of the firm depends on its projected future cash flows. But, as noted above, because of uncertainty, future cash flows are just what accountants normally want to avoid estimating. All accounting systems therefore resort to a similar approach, which may be called the "better to disclose details" rule, for resolving this problem. The problem of valuing the firm is invariably replaced by treating the firm as being composed of many component parts, and listing and valuing the firm's component assets and liabilities. Of course, the value of each component asset or liability still depends on possible future uses/cash flows, but the accountant uses market values of one kind or another to bypass that problem. Market values used include past market values (possibly adjusted for depreciation) and/or estimates of present market values in either a buyer's or seller's market.
For whatever system of accounting, if there are reasonably well-defined values for each of the component assets, the accountant seizes upon them and uses them to provide a "book value" of the firm. If not, the component assets are themselves disaggregated into sub-component parts, and valuation of those sub-component parts is attempted. This disaggregation process continues until eventually all sub-component parts have values assigned, or the assets are considered too subjective to value, and the job is done. Assets like the future benefits from having an active, enthusiastic, management team will be considered so subjective that values are never assigned and the asset will never appear on the accountant’s balance sheet.

Accounting systems are remarkably consistent in this respect. They all attempt to indicate the value of the firm by disaggregating, then disclosing the details of values of separate assets\(^5\). In the process, they all ignore hard-to-value assets. Finally, if there are too many details to fit on, say, one page, the values of like assets are added together. This aggregation is not, however, fundamental to the valuation process. It merely facilitates presentation of summary information.

Note that the "better to disclose details" valuation algorithm normally terminates as soon as an acceptable value, possibly for a group of assets, is found. For instance, a machine can be thought of as an assembly of spare parts, but it is not normal to disaggregate machines into parts to find acceptable

\(^5\) It is sometimes argued that Balance Sheets are not valuation statements, that they are a by-product of using double-entry accounting for income measurement, and that the money values listed are merely unexpired costs. However, it is difficult to believe that a listing of unexpired costs is not intended to provide the reader with some indication of values of the assets and liabilities. Larger firms would be expected to list larger numbers, an indication of larger value.
values. Provided workable values like purchase prices are available for each machine as a whole, each machine will normally be regarded as a separate asset. A firm might also own a full set of spare parts for some existing machines, and these spares would probably cost much more than the machine itself. Yet an accountant would have no sense of conflict of valuation principles if he/she valued the spare parts at their (high) exchange prices, and whole machines at their (comparatively low) exchange prices. The general rule for grouping assets is simple: Keep dividing hard-to-value things like whole firms into parts, but do not go any further than is necessary to arrive at some workable values.6

One apparent exception to the normal termination rule for the "better to disclose details" grouping algorithm is when a business is bought as a going concern. In such a case the purchase price is readily available, and is often used for valuation by the purchaser, yet accounting standards for consolidated reporting insist that the values of the individual assets be disclosed in the consolidated company accounts. Why not treat the acquired business like any other asset? The answer is that whole firms are not like other assets. Whole firms may borrow more money, invest in new assets, dispose of old ones, lend money to their parent, and so on. Within a short time of purchase, the historical purchase price of a firm may bear little relationship to its detailed asset and liability profile. So once again, it is considered "better to disclose details".8

6 MacNeal [1939] mentions the grouping problem and concludes that there is no infallible test. He suggests the "safest guide to follow is custom"[p.171].

7 The asset disclosed would be "Investment, at cost" in the purchaser's books.

8 There are also important tax and legal reasons for continuing to maintain the book values of component assets of even wholly-owned subsidiaries, but these are a consequence of the accounting valuation process, not the cause.
However, this gives rise to an awkward period of transition, during which time there is a conflict between the obvious relevance of the recent purchase price of the firm\(^9\) and the "better to disclose details" rule. Accountants have invented the concept of "goodwill" to reconcile the two methods of valuation. The notion of goodwill, and the problems of accounting for it, stem entirely from accountants' preferences for disclosing details of individual asset and liability values rather than the (market) value of the firm as a whole.

2.3.2 Measurement Error

An important, but rarely mentioned, problem for the accountant is measurement error. Hendriksen [1977, pp.130,132] devotes two pages to problems of verifiability and freedom from bias. Ijiri [1966] suggests that reliability be measured by mean-square-error of a number of different valuations. McDonald [1968] used a research design based on analysis of variance to show that second-hand market values of used cars (roughly equivalent to net realizable values) were more objective than valuations based on conventional depreciation techniques. However, practical accountants pay very little attention to measurement error: one never sees a published income statement with confidence intervals.

Measurement error is particularly important when it comes to measuring income. Consider the case where income is defined as the difference between

\(^9\) If past market prices are relevant for other assets, then the recent past market price for a whole firm (adjusted for material changes since acquisition) is also relevant.
wealth measures at two points in time. Parratt [1961, pp.109-114] analyzes the general case where an indirect measure, \( u \), is to be derived from two other measures, \( x \) and \( y \), each estimated as the mean of a number of observations, each estimate having independent error terms, assumed normally distributed. For any such function, \( u = u(x,y) \), Parratt derives the following equation for estimating the standard deviation, \( s_u \), from estimates of the standard deviations of \( x \) and \( y \), \( s_x \) and \( s_y \) respectively:

\[
s_u = \sqrt{\left( \frac{\partial u}{\partial x} \right)^2 s_x^2 + \left( \frac{\partial u}{\partial y} \right)^2 s_y^2}
\]

Applying the above formula to the measurement of accounting income, where \( \text{Income} = W_{\text{end}} - W_{\text{start}} \), the partial derivatives are 1 and -1 respectively, and the estimated standard deviation of Income, \( s_I \), is

\[
s_I = \sqrt{s_{W_{\text{end}}}^2 + s_{W_{\text{start}}}^2}
\]

To see the implications of the latter equation, suppose that estimation errors are normally distributed and one is 95% confident that (a) the net sum of the values of all assets at time \( t=0 \) is \( W_0 \pm 2s_{W0} \), i.e., plus or minus two standard deviations, and (b) the net sum of values at time \( t=1 \) is \( W_1 \pm 2s_{W1} \). Then, if the error terms are independent, and income is defined as the change in firm value over a period, one would be 95% confident that the firm's income is

\[
(W_1 - W_0) \pm 2\sqrt{s_{W1}^2 + s_{W0}^2}
\]
Further suppose that return on investment is around 10% p.a., with no dividends or other distributions, i.e., $W_1 = 1.1 \times W_0$, and the uncertainty of valuation of assets at both the start and end of the year is also ±10% (i.e., $2\sigma_{W_0} = 0.1 \times W_0$, and $2\sigma_{W_1} = 0.1 \times W_1 = 0.1 \times 1.1 \times W_0$). Income, together with 95% confidence limits, should be reported as $0.1 \times W_0 \pm 0.15 \times W_0$. The 95% confidence limits are 150% of the income estimate! Worse still, the relative uncertainty gets larger as the accounting period gets smaller.

Evidently accountants who wish to reduce measurement error in reported income should not base income measurement on independent start-of-period and end-of-period appraisals of asset and liability values. An example where semi-independent measures were tried and subsequently rejected was Reserve Recognition Accounting. In 1978, the SEC experimented with reporting income based on the difference between the value of proved oil reserves from year to year. In 1981, after numerous studies which found that revisions to estimates of reserves and oil prices (i.e., uncertainty in valuation estimates) dominated the "bottom line", the SEC abandoned plans to use RRA in the primary accounts. This is entirely consistent with the range of errors predicted in the previous paragraph.

Another example of period-to-period-induced measurement error is the valuation of long-term foreign currency liabilities. Ideally the value of a foreign liability is the present value of the future repayments of interest and principal based on the exchange rates prevailing at those future dates. Present

\[ 2\sigma_1 = 2 \times W_0 \times [(0.11/2)^2 + (0.1/2)^2]^{1/2} = 2 \times W_0 \times 0.07433 = 0.15W_0. \]
practice, as represented by AAS 20 [ASA, 1985], effectively estimates the value of the liability at balance day by (i) assuming that today's exchange rate is the best estimator of the future rates, and (ii) discounting the future cash flows at the nominal interest rate. Changes in value from one balance date to the next are included in the income measure. This pragmatic procedure probably leads to reasonable estimates of the ideal, unknowable, value of the liability for the balance sheet. But by including changes of "value" between balance date and the next in reported income, AAS 20 adds to the measurement error (variance) in reported income.

The income measurement problem is analogous to the problem of measuring the difference in speed between two cars going at similar speeds down the same highway. One way of calculating the difference is to stand on the roadside with a radar gun, make independent measures of the speeds of the cars, then to subtract one measure from the other. As shown above, the measurement error would probably be larger than the estimated difference in speed. However, if the cars are close enough together to allow the speed of one to be measured relative to the speed of the other, i.e., with the radar gun inside one of the moving cars, a much more reliable measure of the difference in speed would be obtained.

Thus accounting systems that measure relative changes in position are likely to have lower error than systems that calculate the difference between independent measures of net asset values. From this perspective it is encouraging to note that income measurement techniques like matching revenue and expenses do measure relative changes in position. But, even under this
approach, there are still considerable uncertainties about some of the estimates of relative changes of position. One example is the choice of depreciation expense: the range of possible values is an indication of uncertainty in the resultant income measure.

The solution to the problem of measurement error is not necessarily to fall back on historical cost. In many cases it is still better to report current values of assets and liabilities, even with measurement error, than to go on reporting historical values. The massive foreign exchange losses disclosed by some Australian firms in 1986-87, are a good example where reporting current values of liabilities provides more useful information than historic values. The problem is not that the measurement error exists; it is unavoidable. The problem is that many accountants behave as if measurement error does not exist. Baxter [1984] has argued that it is better to be approximately right than precisely wrong, but comments from practitioners often tell a different story. Comments that current costs are "meaningless numbers", that "lack consistency" [Seed, 1978, p.119], suggest that many people prefer to be "precisely wrong", or at least, not to think about measurement problems at all.

Perhaps the last paragraph is too simplistic. Positive accounting theorists would look for alternative explanations of accounting practice. Two plausible alternative explanations of accountants failure to provide estimates of measurement error are first, that accountants know that measurement errors, large as they are, are potentially swamped by deliberate misrepresentation, and second, that the legal system requires accountants to produce "a number" for
decisions about how much money should go to parties with conflicting economic interests.

A graphic example of the problem of misrepresentation was given by Mr. R.J Schoer, Head of the Companies Division of the Australian Stock Exchange, in an October, 1990 seminar on problems of enforcement of the Exchange's listing rules. According to Schoer, his Division had recently investigated a case where the wife of the Managing Director of an Australian company had purchased a block of land adjacent to her husband's company's factory for $100,000. Not long after, she had sold the same land, via an intermediary, to her husband's company for $40,000,000. Schoer was particularly upset because the transaction was conducted with full knowledge of the company's auditors, who owned the shelf company used as the intermediary! There is no problem with measurement error in the above example, the figures are clear enough. It may even be that in the eyes of the company the land was worth many millions. The problem is the hidden wealth transfer from the company's shareholders to the Managing Director that is not disclosed in the company's accounts. It could be that, because of their knowledge of deliberate manipulation and misrepresentation, accountants see no point in concerning themselves with measurement errors in the actual figures that are disclosed.

Another explanation for apparent lack of concern with measurement error is the need for "a number" for legal settlements. The 1912 testimony of G.O. May In the Matter of the Estate of E.P. Hatch, Deceased, illustrates the problem. It seems that Hatch's beneficiaries could not agree about their entitlements, and May, of Price Waterhouse & Co., had been asked to value a company owned by
the Estate. May had previously certified that a balance sheet which showed Shareholder's Funds to be $486,000 "correctly sets forth the financial condition of the company at January, 1910." [May, 1936, p.241]. However, May had just indicated in court that he would have been equally happy to have certified the accounts as correct if Shareholder's Funds had been $250,000 less. The question then asked by the lawyer was as follows [May, 1936, p.242]:

Q. Now, how will you explain the discrepancy? You have certified that when you showed a surplus of four hundred and eighty-six thousand dollars that statement was correct; now you say that if you had certified to a surplus of two hundred and thirty-six thousand dollars, that also would have been correct.

A. Yes, that is --

Q. Now, they would both have been correct?

A. Well, I think they would both have been correct in that they would be both fair expressions of opinion; and that is all a balance sheet ever is.

In this case, it is clear that May, the accountant, recognized the considerable uncertainty about the figures. But it is equally clear that the dispute would only be resolved by agreement on a single number for the value of the company. May was offering the number that he thought most fairly represented the situation. Similarly, it may be that when accountants report single numbers on a balance sheet or income statement they do so with full knowledge of the uncertainty surrounding the figures, but in response to legal requirements, e.g., debt covenants and laws preventing payment of dividends from capital, for a single number.
Implications of Measurement Error for AIS Design:

As noted in Chapter 1, accounting practice is the result of a complex interaction of social and technical problems. Measurement error is a technical problem that should be minimized, if possible, in designing an accounting system. It appears that income measurement based on "matching" will have much less measurement error than income based on independent wealth measures at different points of time (e.g., CoCoa, discussed in Section 2.4 below).

2.3.3 Sources of data for accounting measurement

The third and final question addressed in this section, Section 2.3, is how so-called "individual" assets are to be "valued" for income measurement. Initially, at least, there appear to be a multiplicity of different values an accountant might use. However, as noted in the "better to disclose details" section, all systems of accounting rely on some sort of actual or inferred market value, or exchange value, as their source of reasonably objective information about the value of an asset. In fact, the full range of possible data sources is listed in Table 2.1. Nowhere is it ever seriously suggested that asset valuation should, or could, be based on any other information. From a system designer's perspective this point is important: it provides an upper bound on the data collection task for accounting systems design.

*** Table 2.1 goes approximately here, see next page ***
<table>
<thead>
<tr>
<th></th>
<th>Information used for Valuation in Various Accounting Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>The precise, detailed historical record of the firm's purchases (including purchases of services like labour, advertising, and rent), sales, borrowings, lendings, and capital transactions (share issues including bonuses, and dividend payments).</td>
</tr>
<tr>
<td>(II)</td>
<td>In the case of a manufacturing firm, the precise historical record of various factor inputs and outputs of each production process.</td>
</tr>
<tr>
<td>(III)</td>
<td>Approximate knowledge of the expected lives, maintenance requirements, and expected residual values of long-lived assets.</td>
</tr>
<tr>
<td>(IV)</td>
<td>Approximate knowledge of current and historical market prices (both buying and selling prices) of all goods and services, (including long-lived assets and semi-finished manufactured goods), foreign currencies, and long-lived liabilities. (Knowledge of past prices means that specific and general price index series are also available.)</td>
</tr>
<tr>
<td>(V)</td>
<td>Approximate knowledge of operating capacities of productive assets, and the economies that potential replacement assets may offer in increasing or reducing consumption of other resources in a production process.</td>
</tr>
</tbody>
</table>
The most readily accessible market value is likely to be the historical cost -- the amount actually paid for the asset in question -- which is automatically captured by the accountability reporting system. Because the HC value is so readily available, and so clearly important on the date of acquisition, it is not surprising that historical-cost accounting has been used so widely, for so long. However, all the other information in Table 2.1 could be collected, if so desired. The only question is whether the benefits of so doing would justify the costs.

2.4 Descriptions of the Major Accounting System Proposals

All the major accounting system proposals are based around an income measure of some sort, coupled with balance sheet disclosure of details of asset and liability values. Whether one measures income by matching expenses against revenue, or from a wealth-based perspective, income measurement under an articulated double-entry accounting system can be thought of as depending on two things: an asset valuation system, and a concept of capital maintenance. The major normative valuation systems identified in this survey are summarized in Table 2.2. The major approaches to capital maintenance are summarized in Table 2.3.

*** Tables 2.2 and 2.3 go approximately here, see next pages ***

When one of the five valuation systems from Table 2.2 is combined with one of the three capital maintenance concepts from Table 2.3, the result is a system of accounting. However, not all 15 combinations make sense. Five of the more important systems of accounting proposed in the literature, roughly
Table 2.2 Normative accounting valuation systems

<table>
<thead>
<tr>
<th>Basis of Valuation</th>
<th>Prime Advocates</th>
<th>Sources of valuation information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group 1: Based primarily on Historical Cost (less expensive data collection)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Historical cost (HC)</td>
<td>Pacioli [1496]</td>
<td>Internal records of market price at date of acquisition. Arbitrary depreciation rules. LIFO or FIFO inventory flow assumptions needed to value Cost of Sales.</td>
</tr>
<tr>
<td></td>
<td>Sanders, Hatfield and Moore [1938]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Littleton [1952a, b]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ijiri [1969, 1975]</td>
<td></td>
</tr>
<tr>
<td>Real HC, adjusted for changes in purchasing power (RHC)</td>
<td>Schmalenbach [1921]</td>
<td>HC restated for general inflation. Needs external general price index. (In the case of the German authors in the 1920s, restatement was back to the pre-World War 1 gold standard.)</td>
</tr>
<tr>
<td></td>
<td>Mahlberg [1923a,b]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sweeney [1928; 1936]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Study Group on Business Income [1952]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AICPA: ARS6 [1963]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AICPA: APB3 [1969]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ijiri [1981]</td>
<td></td>
</tr>
<tr>
<td><strong>Group 2: Specific current market values are required (more expensive data)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replacement cost (RC)</td>
<td>Schmidt [1924]</td>
<td>Need to estimate cost of modern equivalent asset, taking into account obsolescence and operating economies.</td>
</tr>
<tr>
<td></td>
<td>Sweeney [1936]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mathews [1958]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edwards and Bell [1961]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gynther [1966]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Revsine [1973]</td>
<td></td>
</tr>
<tr>
<td>Cash and Cash Equivalent (NRV)</td>
<td>MacNeal [1939]</td>
<td>Cash equivalent is Net Realizable Value (NRV) from orderly disposal of assets.</td>
</tr>
<tr>
<td></td>
<td>Chambers [1966]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sterling [1970]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lee [1984]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wright [1964, 1970]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Solomons [1966]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Baxter [1975, 1984]</td>
<td></td>
</tr>
</tbody>
</table>
Table 2.3 Approaches to Capital Maintenance for Income Determination

<table>
<thead>
<tr>
<th>Money capital</th>
<th>Income is calculated after maintaining the nominal money capital of the firm at the start of the period. Money capital maintenance is hard to defend since it ignores changing purchasing power of money, e.g., due to inflation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purchasing power capital</td>
<td>Income is calculated after maintaining the purchasing power of capital of the firm at the start of the period. This recognizes the impact of general inflation on financial capital, and implies that an investor's main concern is to at least maintain the generalized purchasing power of his/her investments.</td>
</tr>
<tr>
<td>Operating capacity</td>
<td>Income is calculated after maintaining the operating capacity of the firm at the start of the period. This assumes that a firm must maintain its productive capacity intact before it can recognize profits.</td>
</tr>
</tbody>
</table>
corresponding to the five valuation systems shown in Table 2.2, are discussed in the five sub-sections 2.4.1 through 2.4.5 below.

2.4.1 Historical-cost accounting (HC) and Accounting Practice (GAAP)

Pure historical-cost (HC) systems combine HC asset valuation with money capital maintenance. They rely only on knowledge of types (I), (II) and (III) from the Table 2.1 to value assets, and by inference, to calculate income. As Sterling pointed out in his "Defence of Accounting in the United States" [1966] there are many thousands of combinations of rules that an accountant using HC may choose from, e.g., depreciation, standard cost of manufactured goods, direct cost of manufactured goods. As a result, HC accounting systems can generate many different numbers to represent a firm's income for a period.

GAAP systems allow pragmatic departures from pure HC. In Australia, non-current assets may be revalued (AAS 1, para.18 [ASA, 1973], AAS 10 [ASA, 1981]). Revalued assets must then be depreciated on the revalued amount (AAS 4, para.29 [ASA, 1974]) so the capital maintenance concept is no longer money capital. Depreciation charges could well exceed the money cost of a fixed asset. Also, in most countries, inventory values are often based on a "lower of cost and market" rule, which is also a departure from historical cost. In the US, LIFO is used for inventory valuation to achieve "proper matching" for income determination. Irrespective of the variations, however, the emphasis in income

Type III information is used for depreciation and amortization.
measurement in HC and GAAP accounting is on matching revenues and past costs.

Comment: HC and GAAP accounting systems have the simplest data requirements of the systems reviewed in this section, and are sometimes claimed to be the most objective. Their two great weaknesses are, first, their penchant for arbitrarily allocating costs [Thomas, 1969], and second, that they rarely recognize that prices of goods and services are always changing. Probably the latter is more serious. According to Baxter:

Mankind has been astonishingly reluctant to recognize that money is not stable. We have been slow to profit from experience, and quick to forget; men of high intelligence have often been no wiser than the fools. Except when some painful lesson is fresh in our minds, we have been apt to plan our affairs on the assumption that money has a fixed value - and to resent any contrary suggestion. [1984, p.7]

Chambers puts the case against historical-cost accounting more bluntly:

As a method of accounting for inflation, historical cost accounting is universally acknowledged to be defective. [1980, p.134].

In countries like Brazil and Israel, where inflation has been very high, historical-cost accounting is not used. Brazil, for instance, relies on a variant of CPP/CCA accounting [Tweedie and Whittington, 1984, pp.237-240] that is described in more detail in Chapter 6.

---

12 As indicated in the examples in the previous paragraph, some GAAP systems recognize some price changes.
2.4.2 Constant-purchasing-power accounting (CPP)

Constant-purchasing-power systems combine real-historical-cost (RHC) asset valuation with general-purchasing-power capital maintenance. They rely on knowledge of types (I), (II), (III), and the general-price-level-index series from (IV) of Table 2.1. The emphasis in income measurement in CPP accounting is still on matching revenues and past costs, although this time both are expressed in units of constant purchasing power.

**Comment:** If one believes that historical costs are important (presumably because of the objectivity of the figures on the original invoices), and one also wants to recognize that inflation (or deflation) changes the generalized purchasing power of one's unit of account, then CPP accounting makes sense. CPP accounting restates all expenses, including depreciation, into units of (usually) today's purchasing power, and ensures that profit is not recognized until the purchasing power of contributed capital has been maintained. Like HC/GAAP accounting it suffers from arbitrary allocation of costs to different time periods.

For many years until the mid-1970s, CPP accounting was favoured by most accountants around the world as the best method of dealing with inflation. Early work on CPP accounting in the US seems to have commenced with Middleditch [1918], who was, in turn, influenced by Irving Fisher's *The Purchasing Power of Money* [1911]. Tweedie and Whittington [1984, p.19] describe Sweeney as "the great pioneer of CPP accounting in the English-speaking world". After fifteen papers from 1927 onwards, Sweeney published his
doctoral dissertation on Stabilized Accounting [1936] in 1936, and many of his recommendations would be described today as CPP accounting\textsuperscript{13}. In the 1950's and 60's, various AAA committees recommended the use of CPP accounting [AAA 1951; 1957; 1966]. Likewise, the AICPA favoured CPP as the best technique for inflation accounting. In 1952, an AICPA\textsuperscript{14}-sponsored Study Group on changing concepts of income favoured the use of "units of approximately equal purchasing power" [Study Group on Business Income, 1952, para. 12, p.105]. This preference continued in the US professional bodies through ARS No. 6 [AICPA, 1963], APB Statement No. 3 [AICPA, 1969], and the FASB Exposure Draft "Financial Reporting in units of General Purchasing Power" [FASB, 1974]. The British profession also favoured CPP accounting until at least 1973, when Exposure Draft ED 8 "Accounting for Changes in the Purchasing Power of Money" was issued [ASC, 1973]. In Australia, a preliminary exposure draft, "A Method of Accounting for Changes in the Purchasing Power of Money" was issued in December, 1974. Canada issued a similar exposure draft the next year, in June 1975.

Some writers still favour CPP accounting as the single best method of accounting. For instance, Ijiri, one of the staunchest allies of historical-cost accounting, has argued for years [1967, 1975, 1981] that accountability, not decision-usefulness, is the most important objective of accounting. He therefore favours CPP accounting (which he describes as HC accounting in units

\textsuperscript{13} A recent study by Clarke and Dean [1989] suggests that Sweeney [1936] was the earliest English-speaking advocate of "real CCA" [p.303], but many of his proposals seem to advocate CPP accounting.

\textsuperscript{14} Then called the American Institute of Accountants (AIA).
of constant purchasing power) [1981] over the current-value systems discussed below.

2.4.3 Current-Cost Accounting (CCA)

Current-cost Accounting (CCA) systems combine replacement-cost (RC) asset valuation with operating-capacity capital maintenance. They use Table 2.1 type (II), (IV), and (V) data for estimating replacement costs of "current equivalent assets". Changes in the values of assets and liabilities held during a period are included in the wealth measure (balance sheet), but because of the capital-maintenance assumption, are not included in the income measure. Operating income is calculated after deducting expenses measured at replacement cost.

Comment: For some reason, in the mid-1970s, all the professional accounting bodies in the English-speaking world switched their allegiances from CPP accounting to Current-Cost Accounting. In September, 1975, the Sandilands Committee [1975] in Britain recommended current-cost accounting. Across the Atlantic six months later, Burton, the Chief Accountant of the SEC, was influential in bringing replacement-cost accounting into the limelight through ASR 190 [SEC, 1976]. Burton believed that the proper figures to disclose were the economic costs of the firm's assets, and that CPP accounting would not achieve any significant benefit because it did not require any new economic measurements to be made. Within five years there were standards in place in Britain [ASC, SSAP 16, 1980] and the US [FASB, SFAS 33, 1979] requiring mandatory disclosure of CCA information by large firms. Standards in other
countries followed. In Australia, SAP1 [ASA, 1983] recommended full CCA Profit and Loss statements and Balance Sheets, though disclosure was optional. Tweedie and Whittington [1984, Tables 10.2, and 10.3] indicate that many other countries also issued CCA standards or exposure drafts about the same time.

Current-cost/Replacement-cost accounting was not, of course, invented in the 1970s. Schmalenbach [1908] recommended the use of replacement costs for pricing purposes [Clarke and Dean, 1989]. In a 1927 submission on railroad depreciation to the US Interstate Commerce Commission, George O. May (for 25 years the Senior Partner of Price, Waterhouse & Co.) argued that replacement costs should continue to be used in railroad accounts in preference to "original" cost:

> The economic arguments in favor of charging the cost of replacement rather than the cost of the original unit against operating expenses, find ample support in practice; ... it is the practice which has been required by the Commission under existing regulations for the last twenty years in respect to important classes of property, such as track material." [May, 1936, p.180]

Sweeney [1936] argued that ideally, replacement costs should be used in stabilized accounting. And MacNeal [1939] devotes 16 pages to replacement costs (pp.155-171), arguing that it "constitutes the best available index of market value in the absence of an acceptable market price" [p.172]. In later years, Mathews and Grant [1957], Edwards and Bell [1961], Mathews [1965], Gynther [1966], Revsine [1973], and others, also argued that replacement costs are often more relevant to decision-makers than either historical-costs or price-level-restated-historical costs. However, this enthusiasm for CCA by some
theorists and the professional accounting bodies in the late 1970's has not been shared by investors, analysts, and the business world at large. Within a year of promulgation of ASR 190, the US Financial Executives Institute commissioned a survey of executive and user reactions to ASR 190. Two representative comments are as follows [Seed, 1978, p.118]:

(1) No use. Security analysts do not understand this information. No one asks about it.

(2) Not useful and could be misleading. Tells us what it would cost to replace assets that will not be replaced.

Reactions to the 1979/80 US and UK standards that followed ASR 190 and Sandilands and were also negative. So much so, that the mandatory status of both SFAS 33 and SSAP 16 has now been reduced to ineffectual "encouragement" that firms should disclose the effects of changing prices in supplementary statements.

There are two obvious problems with pure replacement-cost CCA. The first is suggested by the second quotation above. If management would not willingly replace an asset at current replacement costs, the replacement cost does not seem relevant; management and investor confidence in CCA suffers accordingly. As discussed in more detail in Section 2.4.5, a solution to this problem seems to be to use "opportunity value" or "deprival value". The second problem is that not everyone wants to maintain operating capacity. Carsberg [1984b] makes this point very strongly in the following comment about operating capacity maintenance as required by SSAP 16 [ASC, 1980] in the UK:
One recommendation from the research stands out clearly. Some companies have strong reasons for arguing that the concept of maintaining operating capability does not apply to them. They do not try to maintain operating capability, they cannot do so, and they would not know if they had. Forcing all companies to make this capital maintenance assumption is causing antagonism to the standard. [p.147]

Under a strict interpretation of CCA, an investor holding a portfolio of securities on a rising market would not report profits; a farmer holding land near a rapidly-growing city would not report profits; and a firm that had bought an asset, like a ship, during a depressed market, and is now earning good cash flows because of an increased demand for the asset's services, might find it has to report losses (prices for new assets now being so high that replacement of operating capacity exceeds the opportunity value of the existing asset). Anomalies such as these suggest that maintenance of operating capacity is not descriptive of what is normally meant by capital maintenance. On the other hand, financial capital maintenance, which restates contributed capital for changes in the general level of prices, seems to more accurately describe the concept that people have in mind when they compare well-offness at two periods of time.

2.4.4 Cash-and-Cash-Equivalent (CCE) Accounting

Cash-and-Cash-Equivalent accounting combines net realizable values for assets with purchasing-power capital maintenance. It uses only Table 2.1 type (IV) data for asset valuation on the grounds that if one is holding an asset the only value relevant to one's decision-making is its disposal value, not its original cost.
or the cost of replacing it. The emphasis in income determination under this approach is on valuing assets, not matching expenses and revenues.

**Comment:** The champions of CCE accounting are Chambers [1966, 1980], Sterling [1970], Lee [1984], and in a sense, MacNeal. In 1939 MacNeal argued that financial statements only display "the truth" when they disclose the current value of assets. MacNeal [1939] pointed to corporations that "with prosperous-looking balance sheets were slapped into bankruptcy" during the depression (when real estate values fell well below the amounts for which they had been mortgaged), and to the potential for management to manipulate investment trust profits by realizing profits when it suited them.

Chambers too, argues, quite compellingly, that "all past prices are simply a matter of history. Only present prices have any bearing on the choice of an action." He goes on to say:

Excluding all past prices, there are only two prices which could be used to measure the monetary equivalent of any nonmonetary good in possession: the buying price and the selling price. But the buying price, or replacement price, does not indicate capacity, on the basis of present holdings, to go into the market with cash for the purpose of adapting oneself to contemporary conditions, whereas the selling price does. We propose, therefore, that the single financial property which is uniformly relevant at a point of time for all possible future actions in markets is the market selling price or realizable price of any or all goods held. Realizable price may be described as **current cash equivalent.** [1966, pp.91-92]

The usual concern with exit price proposals is that the values reported are not generally seen as relevant unless the firm is on the brink of liquidation. Of course, when it is, they suddenly become the only relevant numbers. The view
taken in this thesis is that CCE values are worth collecting only if a firm is in
trouble or for non-routine management decision-making. CCE income is also a
problem: it is based on the difference between two "wealth" measures, and as
discussed in Section 2.3.2, it will therefore suffer from large measurement
error.

2.4.5 Current-cost/constant-purchasing-power accounting (CC/CPP)

Current-cost/constant-purchasing-power (CC/CPP) systems combine
opportunity-value asset valuation with purchasing-power capital maintenance.
They use Table 2.1 type (II), (IV), and (V) data for estimating replacement costs
of "current equivalent assets" or net realizable values for assets no longer worth
retaining in the business. There is an unavoidable grey area between these two
values where one must rely on the judgment of the accountant in asset
valuation. For most assets, however, opportunity value is likely to be
replacement cost. Operating income is calculated after deducting opportunity-
value expenses from revenue. However, unlike CCA the generalized purchasing
power of contributed capital (not operating capacity) is maintained. CC/CPP
systems include changes in asset values above and below the general level of
prices as gains and losses, that is, as part of comprehensive income [FASB,
SFAC 3, 1980b, Definition 6].

Comment: There is a growing consensus that if a single asset value other than
historical cost is to be reported, it should be what Wright [1964] calls
"opportunity value" and Baxter [1984] calls "deprival value". Baxter conjures up
a vivid image of deprival value by saying "if a thief threatens to make off with
one of your assets, but offers to refrain if you pay enough, what is the highest
sum he can prise from you?" [1984, p.200]. Wright’s definition is more formal:

the value to a business enterprise of an asset owned by the enterprise is
given by the least costly of the alternatives avoided through having the
asset ... In general, we must not imagine the enterprise to have been
deprived of its asset suddenly and without warning; rather the situation
to be envisaged is one in which management has had every opportunity to
minimise the loss or damage to which absence of the asset might give
rise. [1977, pp.198-9]

Whether one calls this concept "opportunity value" [Wright, 1964], "deprival
value" [Baxter, 1984], "value to the owner" [Bonbright, 1937], [Solomons, 1966],
"value to the firm" [Sandilands, 1975], "value to the enterprise" [Wright, 1977],
"value to the business" [ASC, SSAP 16, 1980], or "current cost or lower
recoverable amount" [FASB, 1986, SFAS 89], this method of valuation now
seems to have majority support amongst inflation accounting theorists. As
noted earlier, replacement cost is not relevant if a firm would not now replace
the asset. Net realizable value is not relevant if a firm is a going concern, and
fully intends to keep an asset. Opportunity value is said to maintain relevance
in both cases because it swings from a maximum of replacement cost through to
a minimum of net realizable value according to management’s intentions for the
asset.

Since opportunity value uses replacement-cost accounting as its upper limit for
asset value, it suffers from all the problems of replacement-cost accounting
[Ashton 1987]. For example, it is difficult to calculate the replacement costs of
assets subject to great technological change. In addition, the sum of the
opportunity values all a firm’s assets may exceed the value of the firm overall.
For instance, Baxter [1984, p.231] gives an example of a "moribund" railway line with many bridges. All bridges must be usable for the train line to be functional. The opportunity value of each bridge is approximately the net present value of the future cash flows from the entire line, because if the bridge collapses, the train service stops. But in that case the sum of opportunity values will be well in excess of the value of the whole business, which seems absurd. This is a case where the "better to disclose details" valuation rule fails; it is foolish to treat each bridge as an asset. It would be better to treat the whole line as a single asset. One is left with the nagging suspicion, however, that there must be other less-obvious cases where the sum of opportunity values would bear little resemblance to the overall value of the business.

CC/CPP accounting seeks to maintain the real value of money capital, i.e., it adjusts money capital for changes in generalized purchasing power. This means that CC/CPP income is not subject to the criticism which Sterling [Sterling and Lemke, 1981], Carsberg [1981, 1984a], and others have levelled at CCA for its maintenance of operating capacity (discussed in Section 2.4.3 above). Baxter, in Britain, has been a strong advocate of CC/CPP accounting, which he describes, variously, as "stabilized accounting" [1984] and a "hybrid of CCA and CPP" [1985].

If inflation in the US and UK ever rose to such a level that there were calls for mandatory inflation accounting reports, it seems likely that both countries would opt for sufficient disclosure to enable CC/CPP accounting income figures to be calculated. In the UK, all the data needed for constructing CC/CPP reports seem to be available in the British exposure draft ED 35 [ASC, 1984].
the US, the latest (non-mandatory) inflation accounting standard SFAS 89 [FASB 1986] provides a good example of how a CC/CPP standard might be framed.

2.5 Summary

In an attempt to determine the characteristics of "good" accounting systems this chapter has explored some of the huge literature on normative accounting theory. The main conclusions are as follows:

First, there is wide agreement that the primary objective of accounting is to provide information useful for making economic decisions. This primary objective may be subclassified into three major objectives, one relating to management accounting, the other two to financial accounting. Those objectives are as follows:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>To provide management with information useful for planning and control.</td>
</tr>
<tr>
<td>2.</td>
<td>To provide external users with information useful for predicting the future cash flows and the earning power of a firm.</td>
</tr>
<tr>
<td>3.</td>
<td>To provide accountability information useful for contract monitoring.</td>
</tr>
</tbody>
</table>

Second, for accountability reasons, an information system designed to meet the three main objectives certainly needs to be able to store a full transaction history. With careful design such a database of facts might also be useful for management planning and control purposes. (Of course, much of the information required for strategic planning will have to be obtained, often on an ad hoc basis, from sources external to the firm.)
Third, computer processing power does not solve the major ongoing problem of accounting, namely performance/income measurement. Income measurement is important to both management and external users, but it is not clear how income should be measured. Although many different methods of income measurement have been proposed, the underlying approach to income and wealth measurement at any time is remarkably consistent. For very practical reasons, all systems disclose the values of many individual categories of assets and liabilities in preference to estimates of future cash flows. Income for a period is then defined as either (i) revenue less expenses (where expenses are based, in part, on changes in recorded asset values), or (ii) some portion of the change in overall net assets during the period (net of any capital contributions or distributions like dividends). It is apparent that there are likely to be quite large measurement errors associated with either method of calculating income, but (possibly for social and legal reasons) measurement errors are usually ignored.

Fourth, data for use in valuation of individual assets include historical costs, allocations based on costs of inputs to manufacturing processes, expected lives of assets, and current buying and selling prices of equivalent assets. Five major systems of accounting that use some or all of these data were described briefly, with some arguments in favour and against them, in Section 2.4. The five major systems were

HC-GAAP   Historical Cost and GAAP Accounting,
CPP       Current Purchasing Power Accounting,
CCA       Current Cost Accounting,
CCE  Cash and Cash Equivalent Accounting, and  
CC/CPP  Current Cost/Current Purchasing Power Accounting.

Of these, the first two use mainly historical transaction data for valuation and are therefore cheaper to operate than the latter three systems. In times of high inflation, e.g., in countries like Brazil, there seems little doubt that CPP would provide more useful information than HC-GAAP. Of the remaining three systems, there appear to be some defects with CCA's capital maintenance concept, but there are quite convincing logical arguments in favour of the last two systems, CCE and CC/CPP, though for different purposes. What is not clear is whether the extra cost of either CCE or CC/CPP can be justified. Do they provide incremental benefits over HC-GAAP and CPP that exceed the additional cost? The purpose of Chapter 3 is to examine the empirical literature in an attempt to answer this question.

Fifth, from a system designer's point of view, all five accounting systems described above have a lot in common. They all require that a detailed record of a firm's assets and obligations be available at report date, and they all apply some set of reasonably well-defined valuation rules to determine the value of each asset or liability at that date. Data used for valuation of individual assets include transaction histories, expected lives of assets, current buying and selling prices of "equivalent" assets, and operating economies created by replacing old assets with more modern equivalents. Data collection for HC-GAAP and CPP systems is relatively straightforward. Data collection for the other systems can only be described as "more challenging". However, because all systems have so much in common, it does seem that it would be possible to design one
computer-based accounting system architecture to accommodate all of the systems investigated.

(11,700 words)
Chapter 3: Empirical Evidence Concerning the Usefulness of the Normative Proposals for Accounting Systems

The previous chapter presented a number of normative proposals for "improved" systems of accounting, and identified performance/income measurement as a major problem of interest to both management and external users. In the last ten years there have been some major experiments with CCA, and to a lesser extent CPP accounting, in various countries, and various organizations. With remarkably few exceptions they have not been the success that normative theorists had predicted.

The purpose of this chapter is to review the evidence on what has been attempted and what has been achieved. This review covers much the same ground as DeBerg and Shriver's comprehensive 32-page study [1987], but it is able to draw on more recent work, UK experience (including Carsberg and Page's [1984] four-volume report on SSAP 16 [ASC, 1980]), and to examine some studies more thoroughly. In particular, this chapter examines evidence on market efficiency, which DeBerg and Shriver did not do, and it probes the statistical analyses of some studies in more depth.

Empirical studies of the usefulness of accounting systems have adopted two main strategies. The first is simply to ask people about their experiences. For instance, some studies have asked preparers how difficult they found it to collect and organize the relevant data. Others have asked users to explain how they use the information once they have it. Studies of this sort give some insight into both the costs and benefits of the system being used. The second strategy uses market-based accounting research techniques to test for an association between stock price...
movements and the accounting numbers being tested. Assuming capital markets are efficient, if there is no relationship between the accounting numbers (e.g., CCA) and stock prices, then, prima facie, the accounting numbers are not useful for predicting future cash flows. Compared to questionnaire-based research, market-based studies give more believable evidence about the usefulness or otherwise of accounting information, but no information about the costs. Most studies have adopted the second strategy, so most of this chapter is devoted to a review of market-based research into the information content of SFAS 33 [FASB 1979] and SSAP 16 [ASC 1980] earnings numbers.

Market-based accounting research always assumes that capital markets quickly impound all new, relevant, publicly-available information in security prices. Without this assumption one could still test for an association between, say, unexpected earnings and cumulative abnormal returns, but it would not be possible to draw any conclusions about the information content, or otherwise, of the accounting earnings number. This chapter therefore begins, in Section 3.1, with a review of recent evidence concerning market efficiency. It raises some doubts about blanket endorsement of the efficient markets hypothesis. Section 3.2 then examines a number of market-based studies in considerable depth. Section 3.3 considers studies that asked people to describe their experiences with different accounting systems. Section 3.4 summarizes the chapter.
3.1 Market Efficiency

Dyckman and Morse introduce their discussion of market efficiency as follows:

Fama [1970] coined the phrase "efficient market" to describe a market with prices that fully reflect information. He further categorized different levels of market efficiency (the weak, semi-strong, and strong forms) based on the type of information involved. Weak-form market efficiency occurs when prices reflect all information embodied in the past price series. Markets are efficient in the semi-strong form when prices reflect all publicly available information. Strong-form market efficiency occurs when prices reflect all information, both public and private. [p.5]

By convention, in the literature, whenever the terms "market efficiency" or "the efficient markets hypothesis (EMH)" are used without qualification, the semi-strong form is intended.

Beaver [1981, Ch.6], Keane [1983], Dyckman and Morse [1986], and Watts and Zimmerman [1986], amongst others, have surveyed the evidence on market efficiency. Broadly, despite various anomalies, market efficiency emerges as a relatively robust description of capital market behaviour. However, there are some disturbing anomalies. Because market efficiency is so widely accepted, most of this section focuses on the anomalies.

3.1.1 Surveys of empirical studies of market efficiency

Tests of weak-form efficiency include (a) studies of serial correlation and runs tests on security prices, and (b) trading strategies based on movements in past prices. The evidence is consistent with weak-form efficiency, i.e., after allowing
for risk and transaction costs you can't make abnormal returns trading on past price information. A group of studies that find some evidence of weak-form inefficiency is discussed in Section 3.1.2(i) below.

Tests of semi-strong efficiency use (a) event-study techniques to test market reaction to the release of unexpected economically significant information, as well as (b) trading strategies based on any publicly available information. There are numerous studies that show a statistically significant link between quarterly, half-yearly, and annual accounting earnings and corresponding cumulative abnormal returns (CARs), but these are only useful as the second step in the argument that accounting numbers are useful. First it must be established, independently of accounting numbers, that market prices reflect all other publicly available information. If that is true, then the link between accounting and stock prices suggests that earnings numbers are useful.

It is difficult to untangle the effects of changes in other variables in studies of dividend changes, dividend ex-dates, bonus issues, switches in accounting policy with cash flow implications (e.g., FIFO to LIFO), and so on, but most studies show fast, unbiased changes in market prices in response to new information. Foster [1986] reports on three examples of market inefficiency, which he labels "the post-earnings announcement anomalies", "the price-earnings ratio anomaly", and "the Briloff phenomenon". More recently, Ou and Penman [1989] report using information in accounting reports to earn significant abnormal returns. These anomalous results are discussed in Sections 3.1.2 (ii) through (v) below.
The evidence on the absolute value of stock prices as indicators of future cash flows is less clear. The EMH predicts that prices fully reflect all publicly available information. However, no one could claim that the revisions in prices during the October 1987 stock market crash were due to revisions in expectations about future dividend streams. What changed were short-run expectations about selling prices. Thus studies that attempt to relate, say, accounting earnings to market prices (not changes in earnings to changes in prices) must be viewed with caution.

So, what do the experts think about the EMH? After a 42 page review examining evidence for and against market efficiency, Beaver came to the following cautious conclusions:

1. The origin of market efficiency with respect to financial information is security analysis. The empirical evidence arose in response to contentions in the financial and accounting communities that the market is inefficient with respect to certain financial statement information. The early evidence was interpreted to be consistent with the contention that security prices respond quickly and in a sophisticated manner to financial statement data. [1981, p.180; 1989, p.170]

2. Efficient market research has a predominantly empirical tradition and largely preceded any formal theoretical treatment of the topic. .. The lack of theoretical developments is one reason why it is difficult to interpret empirical evidence. [1981, p.180; 1989, p.170]

3. [Beaver] takes no position on the efficiency of the market with respect to any specific information system. The nature of the empirical evidence and the interpretation is likely to change over time and be subject to continuing debate and controversy. [1981, p.180; 1989, p.171]

Dyckman and Morse [1986] are less cautious. The second edition of their book, Efficient Capital Markets and Accounting, provides a lengthy review of the evidence for and against market efficiency. In their closing chapter they conclude:
The major research proposal of this book is to treat the EMH as a maintained hypothesis." [p.89]

Watts and Zimmerman [1986, Ch.4] examine five studies that attempt to discriminate between what they call the mechanistic vs. no-effects hypotheses. The mechanistic hypothesis implies that "the stock market is systematically misled by accounting procedures" [p.71], whereas the no-effects hypothesis is consistent with market efficiency. According to Watts and Zimmerman:

Circumstances are identified in which the competing hypotheses yield contradictory predictions and tests are conducted to determine which hypothesis is more consistent with the evidence ... [but] the studies, in aggregate, do not discriminate between the hypotheses... [p.71]

In short, these experts give qualified support to the EMH. Stock markets of the world clearly react quickly and in a "sophisticated" manner [Beaver 1981 p.152] to information from many sources, but no one suggests that market efficiency is an absolute truth. The following studies provide clear evidence of cases where the market has not been efficient.

3.1.2 Well-documented instances of market inefficiency

3.1.2(i) Stock price predictability, or market overreaction

Some recent studies report evidence of weak-form market inefficiency. DeBondt and Thaler (D&T) [1985] hypothesized that stock markets tend to overreact to good news and bad news. Without using any accounting data, they formed portfolios of
NYSE common stocks from 1926 through 1982 based on cumulative returns above market-wide returns in the prior three years. They reported

Consistent with the predictions of the overreaction hypothesis, portfolios of prior "losers" are found to outperform prior "winners". Thirty-six months after portfolio formation, the losing stocks have earned about 25% more than the winners, even though the latter are significantly more risky.

Two years later, DeBondt and Thaler [1987] were able to report that five other groups of researchers had duplicated their results, one group including Fama himself. Fama and French [1988, p.247] report

Our results add to mounting evidence that stock returns are predictable... There is little in the literature that foreshadows our estimates that 25-45 percent of the variation of 3-5 year stock returns is predictable from past returns.

D&T [1987] also investigated the marked "January effect" reported in their earlier paper [1985, p.803, fig. 3], the size effect (reported by Foster, Olsen, and Shevlin [1984] and others), as well as the suggestion that their earlier results had been due to failure to consider changes in risk premiums, i.e., mis-specification of the CAPM model. In their conclusions they report that (a) the winner-loser effect cannot be explained by changes in risk as measured by CAPM betas, (b) the winner-loser effect is not primarily a size effect, and (c) they have no satisfactory explanations for the January effect, rational or otherwise. Finally, D&T [1987] also investigated Fama and French's (F&F) [1988] suggestion that both F&F's and D&T's results might be dominated by prices prior to 1940. They found the same negative autocorrelation, i.e., predictable price changes, using data from 1965 to 1984, so even recent stock prices show signs of weak-form efficiency.

3 - 7
3.1.2(ii) Post-earnings announcement anomalies

According to Foster [1986]

At least ten studies covering different time periods and different securities and using different methodologies reported evidence of unexpected good/bad news portfolios having positive/negative security returns in the three-month periods subsequent to the earnings announcement. [p.396]

In one of the studies Foster mentions, Rendleman, Jones and Latane [1982] examined quarterly earnings announcements over the period 1971-1980. Using a standardized unexpected earnings (SUE) measure, each observation was placed into one of ten SUE categories. Rendleman, Jones and Latane then calculated Cumulative Abnormal Returns in the 90 days following the announcement of earnings, and reported abnormal returns of as much as 3% (in the most positive SUE category). They concluded that their results

are remarkably consistent in suggesting that the market does not assimilate unexpectedly favorable or unfavorable quarterly earnings information by the day of earnings announcement [p.283].

In a later study, Foster, Olsen and Shevlin [1984] examined quarterly earnings announcements from a sample of over 2,000 firms during the period 1974-1981. They, too, reported a post-earnings announcement drift in the days (+1,+60) averaging -3% for firms in the most negative SUE category and +3% for firms in the most positive unexpected earnings category (with drifts for small firms roughly double the drifts for large firms [p.596]). For quarterly earnings announcements, this is an abnormal return of around 12% per annum.
Both these studies provide clear examples of publicly available information not being impounded quickly into market prices. This is not consistent with the EMH.

3.1.2(iii) The price-earnings ratio anomaly

Foster [1986] also draws attention to a second example of market inefficiency, again reported consistently across many studies, where securities with low price-earnings ratios out-perform those with high ratios in the period after the information needed to calculate the ratios is released. For instance, Basu [1983] studied 1300 NYSE firms over the sixteen years 1963-1980. He ranked firms by PE ratio annually, and formed five portfolios based on low to high PE. He reported a "significant relation between [PE] ratios and risk-adjusted returns for NYSE firms" [p.143] in the 12 months subsequent to the formation of the portfolios. Annual abnormal returns were +4% for the low PE portfolios, and -3% for the high PE portfolio. Again, this is not consistent with the EMH.

3.1.2(iv) The "Briloff phenomenon"

Foster's third example of apparent market inefficiency is what he calls the Briloff phenomenon [Foster, 1979, 1986]. Writing for Barron's magazine for 20 years, Briloff has consistently criticized management for publishing misleading accounts. Foster [1986] examined daily security return behaviour of companies criticized by Briloff in 21 articles published from 1968 to 1984. On average, he found a "permanent" 8.1% drop in returns on the day the critique first became available to the capital market. The quick price reaction is consistent with market efficiency,
but as Briloff claims he only uses publicly available information for his analyses, the permanent drop in returns is inconsistent with the EMH.

3.1.2(v) Ou and Penman's use of accounting numbers to "beat" the market

Ou and Penman [1989] set out to construct portfolios of shares that will earn abnormal returns by using accounting numbers to predict the probability of one-year-ahead earnings increases. Using roughly 15,000 observations (seven years of data from 1965-1972 times roughly 2,000 firms) they investigate correlations between 68 different accounting ratios (e.g., change in dividend per share, return on total assets, change in capital expenditure over total assets) and change in earnings per share (less prior four-year drift). Selecting the 18 most significant correlations they use step-wise LOGIT with 11,300 observations (not all 18 variables were available for all firm-years) to estimate coefficients of a linear model to predict the probability, P, of increased earnings per share next year. Classifying 5,800 firm-years into those with a P-value less than 0.4 and those with P-value greater than 0.6 (i.e., ignoring firm-years with P-values between 0.4 and 0.6 and firms that failed during the year of prediction), their model predicts changes in earnings per share with 67% accuracy. This is significantly better than the 50% accuracy expected by chance.

To test the usefulness of their P-values, Ou and Penman simulate zero-net-investment portfolios of firms with fiscal-years ending on 31 December. Firms with P-values greater than 0.6 are "bought" on 1 April following the annual report, and held for two years. Funds to purchase this long half of the portfolio are financed from "sales" of firms with P-values less than 0.4, which are "sold" short.
Cumulative abnormal returns (CARs) are calculated by subtracting returns from an equally-weighted market index from each firm's monthly return (this ignores firm's betas). A two-year holding period is used because Ou and Penman found P-values predicted abnormal returns for up to three years ahead. Average abnormal returns, ignoring transaction costs, for the long and short sides of the portfolio are reported as 0.1256 over two years [Table 6]. Standard deviations of returns from two thousand randomly constructed portfolios show that the mean of 0.1256 is significantly different from zero (p=0.000).

Since there is a possibility that these results are due to inclusion of more small than large firms in the portfolios (size acting as a proxy for risk), Ou and Penman repeat their simulation by calculating ten size-related return indices to use in place of the equally-weighted market index in calculating abnormal returns. This method reduces the abnormal returns for smaller firms. Average return over two years drops from 0.1256 to 0.0702, but is still significantly different to zero (p=0.000) [Table 8]. Ou and Penman indicate some reservations about this size adjustment because they think the size effect may really be due to market inefficiency [p.320].

Summarizing, Ou and Penman say

On the basis of an extensive financial statement analysis we have derived a summary measure from financial statements that predicts future stock returns. ... It appears that this fundamental measure captures equity values that are not reflected in stock prices. [p.327]

[Our evidence] points to limitations in the traditional approach in empirical analysis in accounting of making inferences about accounting numbers on the basis of contemporaneous associations with prices. [p.328]
3.1.3 Summary, Market Efficiency

In short, the surveys reviewed in Section 3.1.1 and the five examples of market inefficiency discussed in Section 3.1.2 present a quite believable picture of players on the stock markets being sometimes quite clever and sometimes not so clever. The more worrying cases are those where the market is deliberately misled, e.g., prior to Briloff's articles, or where it is slow to impound publicly available information into the stock prices, e.g., Ou and Penman. This means that market-reaction studies provide an indication, but not a foolproof test, of information content of accounting reports.

3.2 Empirical Evidence Concerning the Usefulness of CCA and CPP Accounting: Market Reaction Studies

The purpose of this section is to examine nine market-based studies to see if CCA and/or CPP income have information content over and above GAAP income. It may seem that examining nine studies in considerable depth gives disproportionate importance to market-based research, but there are four reasons for an in-depth analysis. First, it must be understood that there are definitional problems and measurement errors in the data, so conclusions drawn from such data are subject to uncertainties not captured in the tests of significance. Second, irrespective of their conclusions, the studies contain some useful information about current value accounting. For instance, the figures in the studies may be used to compare the relative sizes of HC and CCA income, particularly income from continuing operations (IFCO), and their correlations, in a way that theorists discussed in Chapter 2 could only guess at. Third, the data contained in the studies may also
be used for purposes other than those intended by the original researchers. For instance, some of the F-statistics presented below were not calculated by the researchers themselves. Fourth, as this thesis goes on in later chapters to develop a computer-based accounting system explicitly designed to support inflation accounting, it is important that the case against inflation accounting is well understood, and carefully evaluated.

The most obvious methodological problem with all nine studies is data clustering. All except the first of the studies were conducted using data from the period 1980-1984. During this period inflation was falling in both the US and the UK, e.g., in the US the CPI fell from 13% to 4%. Prima facie, one would expect differences between historical cost and inflation accounting, the dependent variable, to rise as inflation, the independent variable, rises. It is difficult for researchers to come to strong conclusions about this relationship when all observations of their independent variable are clustered in a limited range, e.g., from 4% to 13% p.a..

3.2.1 Beaver, Christie, and Griffin (BCG) [1980]

Beaver, Christie, and Griffin (BCG) [1980] tested for market reaction to the announcement of replacement cost disclosures required by ASR 190 [SEC, 1976]. This is the only study of market reaction to the announcement of replacement cost information reviewed in this chapter. No significant reaction was found.

BCG began by noting [pp.130-1] that analysts were able to estimate replacement cost information from publicly available data [Falkenstein and Weil, 1977], so there will only be a market reaction if the data disclosed leads to changes in beliefs
about future cash flows. They went on to say that the link between replacement costs and future cash flows is unclear:

Suppose that the current replacement cost of an item is greater than what was expected prior to disclosure. What will be the impact on the current price of the stock? ... Currently, there appears to be little or no theoretical basis for constructing hypotheses regarding the direction of that effect. [p.131]

This is not really true. If the current replacement cost of either inventory or plant is higher than expected, it is literally true that one cannot tell how future selling and cost prices will move (it may be that selling prices will rise faster than costs so cash flows will increase, or the reverse), but the accounting reports are not dealing with the future. It is possible to report whether margins in the past increased or decreased (was the firm in such a fiercely competitive market that it could not raise prices to match the increase in costs?), and this may be useful in predicting future margins.

For 553 large firms, BCG collected ASR 190 replacement cost data (for fiscal year-ends 31 December, 1976 and 1977) from the SEC. Daily market price information was available from the CRSP file, and GAAP information from the Compustat Annual Industrial Tape. They found "it was difficult to obtain a 'clean' estimate" of the difference between net income under historic cost (HC) and net income under replacement cost (RC) from ASR 190 data. For example, ASR 190 required that the increase in both cost of sales and depreciation be reported, but there is an untraceable common element in these figures, i.e., depreciation for the current year flows into cost of sales through overhead allocation in the cost accounting system.
Pragmatically, BCG calculated the difference between HC and RC income (only for companies that reported HC profits, not losses) as follows:

\[
\text{RC income} = \text{HC income} - (\text{Increase in Cost of Sales + Increase in Depreciation})
\]

Clearly this calculation double-counts depreciation in Cost of Sales; it also ignores holding gains on net monetary liabilities. For 334 firms, BCG calculate that on average Replacement Cost net income would have been 23% lower than historical cost net income. Many US firms use LIFO for inventory valuation, so not surprisingly, the greater part of the decrease was due to depreciation.

Because ASR 190 might have had market-wide effects it was not possible to use Ball and Brown’s methodology to calculate cumulative abnormal returns, i.e., returns of a firm in excess of beta times the return of the market. Instead, BCG partitioned their company data into eight portfolios, two of which are of particular interest. In one comparison (Pair B) BCG compare Portfolio 2, which consisted of those companies where

(a) the reported difference between Historical cost and Replacement cost Cost of Sales was higher than average, AND
(b) the reported difference between Historical cost and Replacement cost Depreciation was higher than average, AND
(c) Net monetary liabilities were lower than average
to Portfolio 7, where all the above were reversed. The betas of each firm were estimated based on five years of monthly data prior to the period of interest, and weightings of high and low beta securities in the portfolios were adjusted so that overall portfolio betas were all equal to one. In these circumstances, one would expect replacement cost income to be much lower than historical cost income for Portfolio 2, but that Portfolio 7 would be very little affected. If this were new information to the stock market the return of portfolio 2 after announcement of the information would drop relative to the return of Portfolio 7. BCG report no significant difference between the returns of the two portfolios.

One interpretation of the above results is that the market had perfectly anticipated the published figures. To test this interpretation BCG used estimates of replacement cost income published by Value Line, a professional analysis company, and partitioned companies into portfolios on the basis of the difference between "actual" replacement cost income and Value Line's estimate. Again, the differences between the returns of the various portfolios were insignificant.

BCG conclude that their findings are consistent with the hypothesis that ASR 190 replacement cost disclosures provided no information to the market during the fifteen trading days before and after the date that the requirement was first proposed, the date that the requirement became effective, and the date that the data were first filed with the SEC. [p.155]

They also comment, however, that their results may have been confounded by other effects, so the inability to reject the null hypothesis may have been due to misspecified research design.
3.2.2 Lustgarten [1982]

Lustgarten [1982] is a clear, clean, simple, regression-based study of market returns for the ten month period 6 months before the first ASR 190 information announcement to 4 months after. The study estimates abnormal returns for the 10 months using the conventional market model approach. It then regresses abnormal returns for 581 companies against three variables:

\[ x_1 \] unanticipated historical cost earnings (expected income calculated assuming a random walk with drift model)

\[ x_2 \] the difference between replacement cost accumulated depreciation and historical cost accumulated depreciation

\[ x_3 \] the logarithm of sales as an indicator of firm size.

Deflating these variables by Earnings, Assets, Number of shares, and Market value, Lustgarten found significant t-statistics on the \( x_2 \) coefficient when \( x_2 \) was deflated by Assets or Number of Shares. To check the regressions he ranked his companies by the \( x_2 \) variable, partitioned them into four portfolios, and calculated Cumulative Average Residuals (CARs) in the normal way. Consistent with the regressions, he found returns for the portfolios with the higher \( x_2 \) values (the highest-\( x_2 \) portfolio consisted mainly of steel, aluminium, railroad, auto, chemical and textile companies) were lower than for the lower \( x_2 \) portfolios. Surprisingly most of the drop in market value took place "five or six months before the end of the company's fiscal year" [p.137]. Lustgarten concludes
The test results above indicate that the ASR 190 disclosure was associated with abnormal returns. ... The larger the excess of replacement depreciation over historical depreciation, the larger the decline in share values in the months prior to disclosure. ... The explanation offered is that publicity surrounding the announcement of ASR 190 stimulated outside production of information which was the same as that generated by the ASR 190 filings.

Lustgarten's study is important because it was one of the first studies to find an association between market prices and replacement cost information. In the light of Bernard and Ruland's [1987] evidence of close correspondence between analysts' estimates and the actual figures disclosed, his explanation is credible. His results, especially Table 3 [p.132], are also consistent with a not over-efficient market taking more interest in replacement cost information after the SEC had made it "officially" important by issuing ASR 190.

3.2.3 Beaver and Landsman (BL) [1983]

In an influential study commissioned by the FASB, Beaver and Landsman (BL) [1983] examined the relationship between annual security returns and annual percentage changes in several earnings variables from SFAS 33 [FASB, 1979]. Unlike the BCG study (Section 3.2.1), which was interested in the information content of CCA announcements, BL wanted to see if the stock market was sensitive to the information of the type measured by current cost income, i.e., whether security returns already "reflect" such information through availability of substitute information. Noting that SFAS 33 data were more comprehensive than ASR 190, and that price change data had been published for more years (so investors and analysts had longer to learn how to use that information) they set out to look for a link between market prices and CCA information.
BL focused on 731 "non-financial" firms (i.e., not banks) with 31 December year-ends and data on the Compustat file. Because of the FASB reporting rules, the firms in the sample were the largest in the US, 95.5% were audited by the "big eight", and they spanned most industries. The eight key variables used in the BL study (using BL terminology) were as shown in Table 3.1.

Table 3.1: Variables used in Beaver and Landsman [1983]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETURN</td>
<td>Annual common-stock dividend plus capital gain divided by start of year stock price. This &quot;raw&quot; RETURN variable includes market-wide price movements. It has not been adjusted via the market model. (Source: Compustat file; &quot;annual&quot; because SFAS 33 data was only available annually.)</td>
</tr>
<tr>
<td>HC</td>
<td>Percentage change in GAAP earnings per share (i.e., % change in earnings available to common shareholders before extraordinary items). (Source: Compustat)</td>
</tr>
<tr>
<td>CF</td>
<td>Percentage change in &quot;cash flow&quot; per share, where &quot;cash flow&quot; is GAAP earnings plus depreciation, amortization and depletion. With LIFO inventory valuation, CF is similar to Cash Flow from Operations per share; it ignores cash outflows for new equipment.</td>
</tr>
<tr>
<td>PRE</td>
<td>Percentage change per share in current cost income from continuing operations, CCIFCO. (Source: SFAS 33 databank)</td>
</tr>
<tr>
<td>CD</td>
<td>Percentage change per share in constant dollar income from continuing operations, CDIFCO. (Source: SFAS 33 databank)</td>
</tr>
<tr>
<td>PREP</td>
<td>PRE plus percentage change per share in purchasing power gain or loss on net monetary liabilities. (based on SFAS 33 databank)</td>
</tr>
<tr>
<td>CDP</td>
<td>CD plus percentage change per share in purchasing power gain or loss on net monetary liabilities. (based on SFAS 33 databank)</td>
</tr>
<tr>
<td>POSTP</td>
<td>PREP in absolute terms (not percentage change) plus holding gains in excess of inflation (&quot;holding gains (HG) on the assets during the year due to changes in the current cost of the assets&quot; less &quot;that portion of HG due to the general increase in prices&quot;) divided by Shareholders Equity at current cost. POSTP is supposed to correspond to the rate of return in real terms. (based on SFAS 33 databank)</td>
</tr>
</tbody>
</table>

In terms of the discussion in Chapter 2, change in CPP income is reflected in CDP, and CC/CPP income (not "change in") is reflected in POSTP. Not surprisingly,
because of the common components in their calculation BL found quite high correlations between some variables [p.54]:

<table>
<thead>
<tr>
<th></th>
<th>1980</th>
<th>1981</th>
</tr>
</thead>
<tbody>
<tr>
<td>HC vs CDP (CPP)</td>
<td>.73</td>
<td>.64</td>
</tr>
<tr>
<td>HC vs PREP</td>
<td>.73</td>
<td>.71</td>
</tr>
<tr>
<td>HC vs POSTP (CC/CPP)</td>
<td>.30</td>
<td>.22</td>
</tr>
</tbody>
</table>

Correlations are important to evaluations of the merits of, say, CCA over HC accounting. If the correlations were high across all firms it would be hard to justify the more expensive system. They are also important in research designs using linear regression, because high correlations between variables makes the coefficients of regression equations unstable. Following from Lustgarten one might expect lower correlations in heavy industries. However, BL did not report correlations classified by industry type.

For the most important part of their study, Beaver and Landsman undertook a cross-sectional (across many companies, all data from one year) study of the relationship between security returns (RETURN) and the other seven variables above. This they justify on the grounds that previous studies "had found a significant, positive cross-sectional correlation between security returns and changes in historical cost earnings" [p.55], and because time-series analysis was not feasible with only two years of changes in earnings. In other words, they had no choice. Using a two-stage regression technique\(^\dagger\) BL regressed RETURN on HC

\(^\dagger\) It has since been shown that BL's two-stage regression approach is equivalent to multiple OLS regression [Christie, Kennelley, King and Schaefer, 1984], [Beaver, 1987]. An example comparing the techniques is included in Appendix 3.1.
and orthogonally versions of the remaining six variables above. They found significant coefficients of the HC variable, but no other explanatory variables with significant coefficients in all three years, 1979-1981. They therefore concluded:

Statement 33 earnings provide no explanatory power with respect to differences in annual security returns across firms over and above that provided by historical cost earnings. [p.73]

This was a damning conclusion. It is, however, instructive to assess BL's findings by examining their figures in more depth. Combining data from Tables 18 and 20 (pp. 62 and 68) it is possible to deduce (i.e., the equations below were not published in the BL study) the equations linking RETURN, HC and POSTP that BL would have obtained had they used multiple regression on the two variables of interest: percentage change in historic cost income (HC), and CCA income (POSTP) (regressions based on change in CCA income cannot be deduced from the tables). The equations, with t-statistics estimating the significance of the coefficients in brackets, are as follows:

\[
\begin{align*}
1979: \quad \text{RETURN} &= \alpha_1 + 0.31^*\text{HC} + 0.35^*\text{POSTP} \quad (R^2 = 0.24) \quad 13.3 \\
& \quad \quad \text{(9.6)} \quad \quad \text{(3.1)} \\
1980: \quad \text{RETURN} &= \alpha_2 + 0.49^*\text{HC} + 0.54^*\text{POSTP} \quad (R^2 = 0.24) \quad 12.4 \\
& \quad \quad (7.9) \quad \quad (3.4) \\
1981: \quad \text{RETURN} &= \alpha_3 + 0.22^*\text{HC} - 0.17^*\text{POSTP} \quad (R^2 = 0.09) \quad 8.9 \\
& \quad \quad (5.2) \quad \quad (-1.0)
\end{align*}
\]

Five of the six t-statistics are significant. This includes 2 of the 3 coefficients of the CCA variable, which happened to be significant in the years of higher inflation.

---

2 CPI percent change figures are from Sharpe [1985, p.10].
But because BL suspect that the regression residuals may be correlated, they comment (sensibly) that "Until a longer time series becomes available, the t-values are to be viewed largely as descriptive statistics rather than to be taken literally." [p.63]

The coefficients of HC and POSTP in the above equations do not indicate the relative importance of the two variables in explaining returns because HC is the % change in earnings per share from one year to the next (e.g., it would be 100 if earnings doubled), whereas POSTP is the CCA real return on equity in one particular year (e.g., 0.20 for a 20% return). To assess the additional information content of the POSTP variable, one uses an F-test on the increment to $R^2$ when the POSTP variable is added to the regression equation. BL did not do this test for just these two variables, HC and POSTP, but the data are available in their Table 19 [p.65], so the relevant F-statistics were calculated (see Appendix 3.2, Panel B). Results are as follows:

<table>
<thead>
<tr>
<th>Year</th>
<th>$R^2$ from regression of RETURN on HC alone</th>
<th>$R^2$ from above</th>
<th>Number of observations</th>
<th>F-value</th>
<th>% CPI change [Sharpe, 1985]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1979</td>
<td>0.22</td>
<td>0.24</td>
<td>392</td>
<td>10 ± 5 *</td>
<td>13.3</td>
</tr>
<tr>
<td>1980</td>
<td>0.21</td>
<td>0.24</td>
<td>323</td>
<td>10 ± 6 *</td>
<td>12.4</td>
</tr>
<tr>
<td>1981</td>
<td>0.08</td>
<td>0.09</td>
<td>297</td>
<td>3 ± 3</td>
<td>8.9</td>
</tr>
</tbody>
</table>

* = F-value significant at the 5% level if greater than 3.9
Since the F-test depends on the difference between the $R^2$s in columns 2 and 3, and the differences are small, the F-value can only be calculated to one significant digit. But even allowing for with this uncertainty the F-values for 1979 and 1980 are significant. Again, BL point out that there may be biases in the F-values if the residuals are not random, but they provide no data for analysis of the residuals.

It is apparent, therefore, that although the CCA data was not significant in all three years, the BL conclusion that "Statement 33 earnings provide no explanatory power with respect to differences in annual security returns across firms over and above that provided by historical cost earnings" is a little strong. It is also apparent, however, that the increase in the $R^2$s when the CCA variable is added to the equations above is small. This means that during the three year studied, CCA income, if significant at all, was much less important than historical cost in explaining stock market returns.

3.2.4 Bublitz, Frecka and McKeown (BFM) [1985]

Bublitz, Frecka and McKeown (BFM) [1985] conducted a more rigorous study that replicated and extended the BL study. It used two years more data than BL and cumulative abnormal returns from April to March (betas based on the prior 60 monthly returns) rather than raw returns over calendar years. (Because most US annual reports are announced in March, their information content should be reflected in stock prices by the end of March.) BFM also separated holding gains into realized and unrealized components, and they took more care of econometric issues such as appropriate deflators, use of adjusted $R^2$s, multi-collinearity, and cross-sectional correlation between industry groups. For the results in Table 4 of
their paper [p.17] they first regressed CARs on historical cost and cash flow variables, then used F-tests to gauge the significance of adding five more SFAS 33 variables. F-values were significant at the 5% level in all four years (1980 - 83).

For the results in their Table 5 [p.19] they used the same technique to show that the change in realized holding gain added significant explanatory power to the change in historical cost income from continuing operations. Other variables, like purchasing power gains and losses considered in the next paragraph, added relatively much less explanatory power.

<table>
<thead>
<tr>
<th>Year</th>
<th>Adjusted R²</th>
<th>Degrees of freedom</th>
<th>Sample size</th>
<th>F-value</th>
<th>%CPI change to previous December</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>before</td>
<td>after</td>
<td>before 7</td>
<td>after</td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td>0.165</td>
<td>0.190</td>
<td>5 7 328</td>
<td>5.98 **</td>
<td>13.3</td>
</tr>
<tr>
<td>1981</td>
<td>0.128</td>
<td>0.143</td>
<td>5 7 350</td>
<td>4.02 **</td>
<td>12.4</td>
</tr>
<tr>
<td>1982</td>
<td>0.014</td>
<td>0.033</td>
<td>5 7 323</td>
<td>4.12 **</td>
<td>8.9</td>
</tr>
<tr>
<td>1983</td>
<td>0.098</td>
<td>0.097</td>
<td>5 7 361</td>
<td>0.80</td>
<td>3.9</td>
</tr>
</tbody>
</table>

** Significant at the 5% level (the F-value for a sample of >120, and two extra degrees of freedom in numerator is 3.07).

As with the BL study it is instructive to make some additional calculations using BFM's data. From their Table 5 one can also calculate that adding change in purchasing power gains and losses also added significantly to the explanatory power of the regressions for 1980, 81, and 82, but not 1983. The table above (again,

3 Realized holding gain = HC income from continuing operations less CCA income from continuing operations. Since many U.S. firms use LIFO for calculating Cost of Sales the difference is largely the increase in depreciation.

4 This is confirmed by Murdoch's findings for the same years, in Section 3.2.7 below. One wonders if the low inflation in the year ended December 1982 is related to the lack of significance in annual reports published in early 1983.
calculated using a spreadsheet as shown in Appendix 3.2, Panel B) shows adjusted \( R^2 \) both before and after adding the two variables used to measure change in purchasing power. The increase in \( R^2 \) is used to calculate the F-statistic.

Even more than BL, BFM were concerned about non-normality of their regression residuals. Exploring these residuals they found different contributions from different industries [Table 6, p.20]. Overall, they summarized their findings as follows:

We conclude, with suitable caveats, that Statement No. 33 disclosures are associated, after controlling for historical cost, with the information set used by the market to establish security prices.

Because of the care taken it this study its findings are the strongest evidence presented in this chapter of an association between CCA-type information and market returns. It is not hard to understand why the "information set" apparently used by the market included components equivalent to realized holding gains (a component of HC income) and purchasing power gains and losses (a component of CPP and CCA income). But, given Lustgarten's finding (Section 3.2.2) of a significant relationship between returns and RC accumulated depreciation, it is difficult to understand why the market should not be sensitive to unrealized holding gains (also a component of CCA income). Perhaps unrealized holding gains are not a useful predictor of future cash flows.
3.2.5 Schaefer [1984]

Schaefer [1984] investigated the relationship between three independent variables he used for portfolio selection

(i) unanticipated current cost income from continuing operations (CCIFCO),
(ii) unanticipated historic cost from continuing operations (HCIFCO), and
(iii) dividends,

and one dependent variable, unexpected portfolio returns. The objective was to compare unexpected returns of the different portfolios, and so find if current cost earnings "provide a basis for determining which dividend signals have implications for firm value and which do not" [p.647]. Schaefer's conclusion was that CCIFCO did not contain significant information beyond that in HCIFCO and dividends. Because of the high correlations between the variables this conclusion is (at least in retrospect) not too difficult to understand. Nonetheless, it raises doubts about Lustgarten's findings (Section 3.2.2), and leaves one wondering if it is worth going to the trouble of calculating current cost depreciation.

3.2.6 Board and Walker [1985]

The preceding studies were all based on US data. Board and Walker [1985] tested London share market reaction to the announcement of CCA income for 163 UK companies reporting SSAP 16 data for the year ended 31 December, 1981. Between 1980 and 1981 average HC earnings per share fell by 0.79p, while average CCA
earnings per share fell by 0.26p. The correlation between the two earnings measures was 0.86. Cumulative abnormal returns (CAR) were calculated using the market model approach, and the association between dependent variable CAR and independent variables HC and CCA was tested using a 2x2 chi-square test and least squares regression. The chi-square test showed both HC and CCA earnings per share were significantly associated with CAR at the 1% significance level. The regressions yielded $R^2$s of .16 for CAR against the HC variable by itself, and .12 for CAR against the CCA variable by itself. In other words, HC had greater explanatory power than CCA. Board and Walker then went on to use a slightly erroneous two-stage regression\(^5\), and found the coefficient on their HC variable was significant. Because the CCA variable is the third in a three-stage regression its significance is unclear, but the coefficient on their CCA variable (Table 5, equation (6)) is so small it was probably not significantly different from zero.

3.2.7 Murdoch [1986]

Murdoch [1986] is a variation from the research designs above. Murdoch seized upon the possible significance of BL's variable POSTP (discussed in Section 3.2.3) as justification for a matched-pairs study (matching firms on the basis of Value Line's industry classification and Value Line's beta) to explore relative changes in security returns. For example, suppose there are two firms, both in the same industry, and both with similar betas. Murdoch tested to see if a difference in the two firm's accounting returns were matched by a corresponding difference in security returns. He defined six difference-in-returns (6) variables:

\[^5\] See Appendix 3.1
\( \delta S \) (price at end - price at start + dividends)/price at start

\( \delta H \) HC income from continuing operations/HC equity

\( \delta CD \) CPP income from continuing operations/ CPP equity

\( \delta CC \) CCA income from continuing operations/CCA equity

\( \delta PP \) purchasing power gain on net monetary items/ CPP equity

\( \delta NH \) net holding gain/CCA equity

The statistical tests commenced by regressing \( \delta S \) against \( \delta H \) and calculating the coefficients of determination (\( R^2 \)). These were 0.08, 0.14, 0.00 respectively, in 1980, 1981 and 1982. Murdoch then added a CCA or CPP variable to the regression (i.e., he regressed \( \delta S \) against variables \( \delta H \) and \( \delta CD \), \( \delta H \) and \( \delta CC \), \( \delta H \) and \( \delta PP \), and \( \delta H \) and \( \delta NH \)) and again calculated the coefficients of determination (\( R^2 \)). These were 0.09, 0.15, 0.01 for \( \delta H \) and \( \delta PP \) in the same three years. Significance of the SFAS 33 data was then determined using an F-test identical in form to that in Appendix 3.2 to this chapter. On the basis of these tests, Murdoch reported that

we are 98.5 percent confident that purchasing power returns add to the explanatory power of historical data over the three year period. [p.283].

It was this conclusion that motivated the extra tests reported in Section 3.2.4 above with the BFM data. As reported above, the findings using BFM data for 1980-82 (the period covered by Murdoch's data) are consistent with Murdoch, though 1983 was not. This leads one to wonder if Murdoch's result is just a chance product of the data he happened to use in his study.
In addition, Murdoch concluded that

Historical cost returns on equity do not possess incremental information content beyond that provided by current cost returns on equity in explaining security returns [p.286].

This conclusion is inconsistent, though not markedly so, with the other studies. Murdoch's current cost return on equity variable uses income from continuing operations (CCIFCO) in the numerator. Schaefer (Section 3.2.5) found CCIFCO had no more explanatory power than HCIFCO, but presumably the reverse was true. On the basis of significance of t-statistics after regressing RETURN against their PRE and HC variables BL (Section 3.2.3) also found HCIFCO added explanatory power. BFM (Section 3.2.4) make no explicit comment about the relative importance of HCIFCO and CCIFCO. They do report correlations between the two variables for 1980 through 1983 were 0.767, 0.701, 0.768, 0.853 [Table 3, pp.14-15], but in their subsequent tables they use the HC variable, not CCIFCO. Presumably, therefore, they found the two variables contained similar information, but HCIFCO had a stronger relationship to CARs.

3.2.8 Peasnell, Skerratt, and Ward [1987]

Peasnell, Skerratt, and Ward (PSW) [1987], in an extension of Skerratt and Thompson [1984], studied market anticipation of CCA earnings for 208 British companies from 1980 to 1984. Reasoning that as the annual report date approaches "marginal gains from further revisions of HC profit forecasts decline; attention will increasingly turn to other determinants of share price, perhaps including CCA earnings" [p.4], PSW sought to detect the impact of forthcoming CCA information
shortly before announcement. They therefore modelled market return for k days
(k=1..35) before the annual earnings announcement as follows:

\[ R_{jk} = a_k + b_k R_{mk} + c_k HCE_j + d_k CCA_j + U_{jk} \quad \text{(1)} \]

where \( R_{jk} \) is the return of share j measured over k days up to and including the
announcement day, \( R_{mk} \) is the corresponding market return, \( HCE_j \) is the unexpected
historical cost earnings (compared to latest consensus analysts forecasts) as a
fraction of the forecast, \( CCA_j \) is change in CCA earnings per share from last year,
and \( U_{jk} \) is an error term. PSW used various measures of CCA earnings, including
the profit attributable to shareholders (i.e., including the CCA gearing adjustment
(Table 2, p.6)), and the results were very similar.

PSW's principal finding was that the coefficients \( b_k \), \( c_k \), and \( d_k \) were all significant
at the 5% level for all regressions from 5 to 35 days before announcement. Taking
the 10-day case as representative, the regression equation for returns up to the
close of business on the day of announcement was as follows (Table 1, p.4, k=10):

<table>
<thead>
<tr>
<th>( R = a + 0.92 R_m + 0.10 HCE + 0.01 CCA )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(7.5) (3.6) (2.99) (t-statistics)</td>
</tr>
<tr>
<td>(300 cases, ( R^2 = 0.21 ))</td>
</tr>
<tr>
<td>(significant at 5%)</td>
</tr>
</tbody>
</table>

In this example the coefficient of \( R_m \) is about 1, as one would expect, and the
average 10-day return was about ten times more responsive to unexpected
historical cost income, HCE, than to unexpected current cost income, CCA. PSW
note that one reason for the difference in magnitude is that the HCE variable was
based on much more recent analyst information than the CCA variable. In the light
of this it is remarkable that the CCA coefficient was significant at all. PSW tested
for collinearity between the unexpected components of these variables and found a very low correlation (0.10).

The above 5- to 35-day regressions focused on short-term returns for the days before the annual earnings figures were announced. Towards the end of their paper PSW also consider longer-term returns. In Table 7 [p.12] they report the results of a regression similar in form to equation (1) above (without the k subscripts), but where the returns and proportionate earnings changes were measured over two years between the 1980 earnings announcement and the 1982 announcement. The resulting equation was as follows:

\[ R = 0.18 + 0.09R_m + 1.02^{*}\text{HCE} + 0.001^{*}\text{CCA} \quad (90 \text{ cases, } R^2 = 0.55) \]

\[
\begin{array}{cccc}
0.12 & 10.22 & 0.22 & (t-\text{statistics}) \\
\end{array}
\]

Only the coefficient of the HC variable (HCE) was significant. PSW note that betas are usually calculated on monthly or more frequent data, so they were not perturbed by the low coefficient of \(R_m\). Nonetheless, considering that the \(R^2\) is so high, it is hard to understand how the returns for the sample cases were so different to the market overall. Perhaps they are not representative of firms generally? In addition, a problem with using proportionate changes in earnings as variables is that if earnings figures in the base year are small (positive or negative) the proportionate changes are likely to be large. Perhaps it is better to deflate changes by large positive numbers like market value of shares. These concerns aside, the last equation suggests that the market is much more sensitive to changes in expectations for HC income than to changes in CCA income.
3.2.9 Bernard and Ruland (BR) [1987]

In contrast to all the cross-sectional studies, Bernard and Ruland (BR) [1987] adopted a time-series research design. They used a procedure similar to Falkenstein and Weil [1977] to estimate current cost from continuing operations, CCIFCO, for 113 firms for 19 years, 1962-1980. SFAS 33 data were used to check their estimation procedure for 1980 and they found quite high correlations with the actual figures reported\(^6\). Defining unexpected income as the difference between this year's figure and last year's, they found the correlation between unexpected current cost income from continuing operations (UCCIFCO) and unexpected historical cost income from continuing operations (UHCIFCO) was .986 for 72 observations in 1980.

BR divided their data into 27 industry groups, with companies equally weighted in each group. They then regressed the 19 years of mean annual stock returns for each group (not unexpected return) against the 19 mean annual UHCIFCO and UCCIFCO figures using an equation of the form:

\[
R_{it} = a_i + b_{1i} * \text{UHC}_{it} + b_{2i} * \text{UCC}_{it} + e_{it}
\]

where \( \text{UHC}_{it} \) and \( \text{UCC}_{it} \) are UHCIFCO and UCCIFCO both deflated by the market value of common stock at the end of the previous year. Since SFAS 33 allowed firms that use LIFO to treat LIFO cost of sales as the current cost of sales, and

\(^6\) They do not explain why they did not validate their estimation procedures against .981, 1982 and 1983, which presumably they could have done. One would be much more confident with their results if they had.
many US firms use LIFO, the main source of difference between HCIFCO and CCIFCO is current cost depreciation. If depreciation is small relative to other expenses, or large but uniform over time, changes of both variables from last year should tend to move together. It is not too surprising, therefore, to learn that BR found the correlation over 19 years between the two variables UHCIFCO and UCCIFCO exceeded 0.78 for all but two industries. In addition, the average correlations between returns and UHC was 0.27, and between returns and UCC was 0.26. BR comment

Clearly, the movement to a time-series framework has not mitigated the high degree of collinearity between UHC and UCC for the vast majority of industries. In those industries, UHC and UCC are nearly equivalent measures of income, so there is little potential for either to offer incremental information content. [p.714]

The BR findings are consistent with all the earlier findings summarized at the end of Section 3.2.7. In view of the BFM (Section 3.2.4) and Murdoch (Section 3.2.7) findings, it is a pity that BR did not attempt a longitudinal study of the information content of purchasing power gains on net monetary liabilities, or holding gains on non-monetary assets. They may have been discouraged by Walther's comment that the Davidson-Weil model over-estimated purchasing power gain by an average of 68.3% [Walther, 1982, p.376]. However, Walther goes on to report that the differences were due mainly to large percentage errors for firms with small purchasing power gains or losses [p.381], so a longitudinal study may still have been possible.
3.3 Empirical Evidence Concerning the Usefulness of Inflation Accounting: Studies of Preparers, Investor, and Analyst Opinions

3.3.1 Opinions of Management

Despite acceptance by a few firms like General Electric, American Standard, and Philips, inflation accounting is not widely used by US management in its decision-making process. DeBerg and Shriver [1987] report that Madison and Radig [1983] surveyed managers of the 229 largest Fortune 500 firms about their internal use of SFAS 33 data. Only 78 useable responses were received (34%), but even from that sample Madison and Radig report an "overwhelming expression concerning the lack of utility of SFAS 33 disclosures, as perceived by senior corporate preparers of such information" [DeBerg and Shriver, p.72]. The data were not being used internally in decisions concerning inventory levels, dividend payout, equipment replacement, sales contracts, or investment purchases. Furthermore, the respondents indicated that the data were not useful to external users because of subjectivity and inconsistency of assumptions. Surprisingly, however, 72% of respondents indicated that some form of SFAS 33 disclosures should continue.

In Britain, SSAP 16 was issued in April 1980, and withdrawn five years later after widespread non-compliance. In the year ended 31 March, 1983, 269 company accounts (25%) were qualified for failure to publish current cost information in accordance with SSAP 16. A year later the accounts of 530 companies, or 50% of those required to publish SSAP 16 figures, were qualified [ICAEW, 1984, p.18]. Reasons given by directors for their failure to disclose current cost information included (a) costs of preparation exceed benefit to shareholders, (b) it would be of
little value to shareholders, (c) it is not necessary when inflation is around 5%. By April 1985, another year later, 70% of companies were no longer complying with SSAP 16 [ICAEW, 1985a, p.4], and in July 1985 SSAP 16 was officially withdrawn [ICAEW, 1985b, p.19].

Pearcy [1984] interviewed management at six large UK companies which must have been specially selected for their interest in CCA. All said they were committed to the need for current cost information in managing the business [p.221]. Management reported some problems with the "modern equivalent asset" concept and with some of the details of SSAP 16 (like the gearing adjustment calculation). At the Board level, Pearcy reports "All of the companies face the problem that their HC results are the focus for comments by investment analysts, financial journalists, and the outside world generally", and "the cover for dividends is likely to be based on CCA profit". Below Board level "CCA depreciation is usually carried down to the product level". However, because the samples were not random, these comments carry very little weight as measures of management interest generally in CCA.

Concerning the difficulty of obtaining CCA replacement costs, Page [1984] reports on case studies of 16 UK firms prepared by 3 firms of chartered accountants. An important factor seems to be management "attitude" to CCA. Of the companies studied, four were "positive", five were "neutral", and seven were "negative". Page reports:

The principal conclusion to be drawn from this project is that availability of data for calculating the replacement cost of assets has not been an extreme problem for companies required to comply with SSAP 16. Where
costs were measured by the use of internal indices or the results of external indices were checked by another means, companies were much more satisfied with the results than when external indices alone were applied. ... companies who were motivated to devote sufficient resources to the assessment of replacement costs seemed to be satisfied that the results were 'reliable' and 'objective'.

There are indications that companies with a negative attitude towards SSAP 16 have prepared the information with the minimum input of effort and expense. ... Perhaps as a result of this, such companies tended to consider the results as 'uncertain' or 'subjective'. To this extent, negative attitudes towards CCA within companies may be self-reinforcing. [pp.207,208]

The comments by Carsberg [1984b] on the reliability of current cost measures in 13 companies tend to support the above comments by Page. It would seem that reasonable estimates of replacement costs are possible, at reasonable cost, if one is motivated to try. Nonetheless, Carsberg notes that there may be some cases, e.g., replacement cost of an oil rig in the North Sea, where it does not make sense to try to estimate replacement costs, and in such cases, he suggests, firms should be allowed to disclose historical cost and explain why replacement costs were meaningless.

3.3.2 Opinions of investment analysts

The March/April, 1983, edition of the Financial Analysts Journal contained two articles on the use of inflation-adjusted accounting data in the US. For the first article, Berliner [1983] sent questionnaires to 500 randomly selected members of the Financial Analysts Federation and received 190 responses (38%). Half did not use the information at all. Only 9% were frequent users. Frequently cited reasons for non-use were (a) the data was not comparable among firms (due to discretion in applying SFAS 33 rules), (b) the data lacked relevance, and (c) the information
was already available from other sources. Analysts were keen, however, to have the details of the principal assumptions and methods used to calculate the disclosures, and to have long-term liabilities reported at market value. For the second article, Norby [1983] interviewed analysts and fund managers to see how they used inflation adjusted information. It seemed that analysts had experimented with using SFAS 33 data in attempting to improve portfolio results, but the data were no longer being used.

There are three studies in Carsberg and Page [1984, Vol.2] concerning the use of CCA information by the Press, Stockbrokers, Stockbrokers reports, and Institutional Investors in Britain. For a semi-randomly selected sample of 58 companies, 649 press articles reporting accounting information were collected from major newspapers during two years, 1982 and 1983. Of these, 90% did not mention CCA figures at all (85% in 1982, 95% by 1983), and 10% mentioned CCA but emphasized HC. Stockbrokers reports presented mainly HC information, but unlike the newspapers, half of the circulars studied at least mentioned CCA information.

Boys and Rutherford [1984] interviewed analyst/fund managers at 13 investment companies to find out how they used accounting information in making investment decisions. They comment:

Although the major purpose of this study was initially to see how current cost accounting was used by institutional analysts, it soon became clear that, in fact, the current cost accounts were not used to any great extent. The focus of attention thus shifted to an examination of why the accounts were not used. [p.116]

The prime purpose of the analytical process for most analysts is to try to forecast future earnings, on an historical cost basis, and thereby determine, generally using an earnings multiple, whether a share is cheap or dear and
whether to buy, hold or sell. At present little use is made of the current cost accounts, except insofar as the dividend cover is computed on a current cost, as well as an historical cost, basis. [p.124]

The findings of Carsberg and Day [1984] in their interviews with 15 analysts employed by stockbrokers tell much the same story. Analysts were asked to evaluate the full annual report of one company. Only half the analysts (8) used CCA information at all. One was very enthusiastic about it. One saw it as "virtually useless". Comments included:

"SSAP 16 gives no information that can't be obtained from elsewhere in the accounts."

"SSAP 16 information is too subjective: 'You can fiddle historic cost profits fairly easily, so lord knows what creative accounting can do with CCA'."

"Clients are not interested."

3.4 Chapter Summary: Empirical Evidence Concerning the Usefulness of the Normative Proposals for Accounting Systems

The two main methodological problems with the nine empirical studies of the CCA experiments analyzed in Section 3.2 are, first, their dependence on the assumption of market efficiency, and second, the unavoidable problem of data clustering, e.g., all US observations of differences between HC and CCA income were made during a period when the independent variable, inflation, ranged between 4 and 14 percent.

Concerning market efficiency, the studies reviewed in Section 3.1.1 and the five examples of market inefficiency discussed in Section 3.1.2 present a quite believable picture of a "savvy", largely efficient market, that responds quickly to
what it understands, learns quickly, is self-correcting over time, but which does not necessarily impound all relevant information into market prices. One imagines that it will not be possible to replicate Ou and Penman's results after 1989 because the market now understands the procedures used, and is impounding the newly understood information into security prices.

The stock market crash of 1987 is further evidence that absolute values of securities are not reliable estimates of the present value of future dividend streams. Keynes' characterization of the stock market as a beauty contest where each investor's objective is not to pick the girl he or she thinks the prettiest but, rather to pick the one that other investors would consider the prettiest, is still very apt. Players in the market want, and apparently very efficiently impound, information that will help them predict how other players will value securities. In such an environment, once any system of beliefs has become established it tends to be self-fulfilling. If management believe that the market responds to historical cost accounting information they will assess the likely profit implications of their major decisions on an historical cost, not current cost, basis. It might be in managements' long-term survival interests to check to see that dividends are covered by alternative profit measures, but to ignore the market's decision-making criteria would not be in their shareholders' best interests, nor their own.

Thus market reaction, or non-reaction, to current cost disclosures is a useful, but not foolproof, guide to the potential relevance of different forms of accounting information.
With respect to the nine empirical studies of CCA and CPP accounting reviewed in Section 3.2, it was found that during the period of study:

(a) For most industries, current cost income from continuing operations was not so different from the historical cost figure. This result does not depend on market efficiency. In both the US, where LIFO is used for inventory valuation in income measurement, and the UK, where FIFO is used:

(i) the correlation between the two measures was high (0.7 or higher) for most industries (e.g., see Bublitz, Frecka, and McKeown [1985, Table 3, pp.14-15]), and

(ii) it would appear that CCA numbers can be predicted quite well from other information.

(b) There is some evidence of a small link between market returns and CCA/CPP information. In the two most rigorous US studies, Lustgarten [1982] found replacement-cost accumulated depreciation was a significant variable in explaining 10-month excess returns, and Bublitz, Frecka and McKeown [1985] found that adding CCA variables added significantly to the explanation of CARs for 300+ large corporations. In another US study, Murdoch [1986] found some correspondence between market prices and purchasing power gains in the years 1980-82. However, this relationship did not persist into 1983, a year of very low inflation. In Britain, Peasnell, Skerratt, and Ward found that in the short run, changes in CC income were
about ten times less important than changes in historical cost income in explaining market returns.

Section 3.3 considered comments from management, analysts, and the press, about the usefulness of CCA, again only during the period 1980-1984. The general conclusion is that while there are a few enthusiasts, the majority of preparers and users did not find CCA information important. It would appear that reasonably reliable, objective, replacement cost information can be prepared at reasonable cost [Carsberg and Page, 1984, Vol. 2, pp.173-176]. It would also appear that senior managements' accounting information needs are very much driven by what the analysts want. In the present environment, where share prices are clearly most sensitive to HC numbers, analysts not surprisingly see little need for CCA information.

Perhaps the most useful conclusion to be drawn from the studies reviewed in this chapter is that in periods of inflation up to, say, 10 to 15 percent (there is no evidence for higher levels of inflation), it would appear that market analysts can probably make reasonable estimates of the CCA numbers from HC reports, and CCA reports do not contain much additional information. (Of course not all accounting reports concern firms whose shares are traded on the world's stock markets, so even at these relatively low levels of inflation there may be a use for CCA reports in other quarters.) This is an important practical conclusion that none of the normative studies reviewed in the previous chapter could address; they lacked the data. The implications of this conclusion for computer-based accounting information system design are discussed more fully in Chapter 6.

(10,100 words)
Appendix 3.1: Problems with Two-stage Regressions: An Example

Problems with two-stage OLS regressions in market-based accounting research were first identified by Christie, Kennelley, King and Schaefer [1984]. Beaver [1987] notes that the problem is quite wide-spread. The following example demonstrates the problems with two-stage regression procedures. The regression using Method 1 is correct. Regressions using Methods 2 and 3 show the results of various erroneous two-stage regressions used by researchers to estimate the dependence of $y$ on $x_1$ and $x_2$. It would have been simpler if they had just used multiple regression. All the two-stage regression equations below should be compared against equation 1(a) to see where they are wrong. All regressions are based on the following data:

Data $x_1$: 112 126 100 114 121 110 111 124 (n=10)  
$x_2$: 5 13 3 7 11 9 8 4 6 2  
y: 79 97 51 65 82 93 81 38 60 86

Method 1: Multiple regression of $y$ on $x_1$ and $x_2$ ($y = \alpha + \beta_1 x_1 + \beta_2 x_2$)

1(a) $y = -124.57 + 1.659 x_1 + 1.439 x_2$  \hspace{1cm} (R$^2 = 0.782$)  \hspace{1cm} ($se y = 10.10$)

Equation 1(a) shows the correct result of a multiple regression of $y$ on $x_1$ and $x_2$. The coefficients and their standard errors (in brackets under the coefficients) are correct according to statistical theory. The $se$ shown under $R^2$ term is $\sqrt{\text{sum(resids}^2/\text{(n-3)})}$, the standard error of the estimate of $y$ (degrees of freedom = n-3).

Method 2: Regression of $y$ on $x_1$ with residuals $r$. then $r$ on $x_2$

The first regression is of $y$ on $x_1$, with residuals, $r$, saved for the second regression. In the second regression the residuals are regressed on $x_2$, with erroneous results.

2(a) $y = -145.1 + 1.927 x_1$  \hspace{1cm} (R$^2 = 0.770$)  \hspace{1cm} ($se y = 10.6$)

$r$: 8.30 -0.67 3.42 -9.55 11.30 4.96 14.16 -15.36 -8.77 -7.81

2(b) $r = -7.85 + 1.155 x_2$  \hspace{1cm} (R$^2 = 0.166$)  \hspace{1cm} ($\beta_2$ AND se of $\beta_2$ are wrong)

In equation 2(b) the coefficient of $x_2$ will always be LESS than that in equation 1(a) [Beaver, 1987, p.142]. Also, the $se$ of the $x_2$ coeff. is low compared to $se$ for $\beta_2$ in 1(a) [Christie, et al., 1984, p.210].
Appendix 3.2: Calculation of F-values from $R^2$ statistics

The general statement of the problem is to calculate the F-statistic due to the increased explanatory power of $q$ additional regression variables, given that $R^2$s (or adjusted $R^2$s) are known for both

Model A: the regression of $y$ against $x_1, x_2, \ldots, x_m$, and
Model B: the regression of $y$ against $x_1, x_2, \ldots, x_m, x_{m+1}, \ldots, x_{m+q}$.

The following three equations are taken from Fogler and Ganapathy [1982, pp. 53-55]:

1. $F_q, (n-m-q) = [(SSE_A - SSE_B)/q] / [(SSE_B/(n-m-q)] \ldots (1)$
   where $SSE$, subscript A or B, is the sum of the squared residuals (errors) from the regressions in model A or B, and $n$ is the number of observations.

2. $R^2 = 1 - SSE/SST \ldots (2)$
   where SST is the sum of the squared differences between each $y$ observation and the mean value for $y$.

3. $R_a^2 = 1 - (n-1)/(n-m) * (1 - R^2) \ldots (3)$
   where $R_a^2$ is $R^2$ adjusted for the degrees of freedom of the error term "used up" when extra explanatory variables are added to the regression equation.

Equations (2) and (3) above may be rearranged to show

\[
\begin{align*}
SSE &= SST * (1 - R^2) \ldots (2a) \\
R^2 &= 1 - (n-m)/(n-1) * (1 - R_a^2) \ldots (3a)
\end{align*}
\]

Substituting (3a) in (2a)

\[
SSE = SST * (n-m)/(n-1) * (1 - R_a^2) \ldots (2b)
\]

Substituting (2a) in (1), and using the fact that SST is common to both regressions, yields (1a). Repeating with (2b) in (1) yields (1b):

\[
\begin{align*}
F_q, (n-m-q) &= [(R^2_B - R^2_A)/q] / [(1 - R^2_B)/(n-m-q)] \quad (1a) \\
F_q, (n-m-q) &=[((n-m)/(n-m-q)(1-R_a^2_B)-(1-R_a^2_A))/q]/[(1-R_a^2_B)/(n-m-q)] \quad (1b)
\end{align*}
\]

These equations are readily evaluated using a spreadsheet. Calculations are given on the next page. Panel A is merely confirmation of the formulae above. Note that Beaver and Landsman [1983] did not use adjusted $R^2$, whereas the later studies did. Panel B contains the F-values reported in the text of the chapter.
## Appendix 3.2 continued. Calculation of F-statistics.

### Panel A: Confirmation of F-values reported in published papers

The purpose of this panel of calculations is **merely to validate** equations (1a) and (1b) for Model's A and B by showing that the calculated F-value, column 9, labelled "Calcd. F-value" is very similar to the value in column 10, taken from the published tables in the respective papers. The final column, column 11, labelled "%error", shows the percentage difference between the calculated and the published F-values. Differences are attributed to use of rounded data in the published R²s.

The only reason for presenting these Panel A figures is to add credibility to the F-values calculated in Panel B. Until one realizes that some studies publish adjusted R²s, and some studies do not, it is impossible to use the data to derive other meaningful statistics.

(a) Beaver and Landsman [1983], Table 19 (adjusted R² not used)

<table>
<thead>
<tr>
<th>Year</th>
<th>Adj-R²_A</th>
<th>Adj-R²_B</th>
<th>R²_A</th>
<th>R²_B</th>
<th>df₁</th>
<th>df₂</th>
<th>obs.</th>
<th>Calcd. BL's</th>
<th>%error</th>
<th>F-value</th>
<th>F-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1979</td>
<td>0.22</td>
<td>0.36</td>
<td>1</td>
<td>4</td>
<td>392</td>
<td></td>
<td></td>
<td>28.29</td>
<td>28.90</td>
<td>2.15</td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td>0.21</td>
<td>0.25</td>
<td>1</td>
<td>8</td>
<td>323</td>
<td></td>
<td></td>
<td>2.40</td>
<td>2.50</td>
<td>4.17</td>
<td></td>
</tr>
<tr>
<td>1981</td>
<td>0.08</td>
<td>0.11</td>
<td>1</td>
<td>8</td>
<td>297</td>
<td></td>
<td></td>
<td>1.39</td>
<td>1.30</td>
<td>-6.59</td>
<td></td>
</tr>
</tbody>
</table>

(b) Bublitz, Frecka and McKeown [1985], table 4, p.17

<table>
<thead>
<tr>
<th>Year</th>
<th>Adj-R²_A</th>
<th>Adj-R²_B</th>
<th>R²_A</th>
<th>R²_B</th>
<th>df₁</th>
<th>df₂</th>
<th>obs.</th>
<th>Calcd. BFK's</th>
<th>%error</th>
<th>F-value</th>
<th>F-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>0.134</td>
<td>0.209</td>
<td>0.136</td>
<td>0.223</td>
<td>2</td>
<td>7</td>
<td>328</td>
<td>7.18</td>
<td>7.22</td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>1981</td>
<td>0.163</td>
<td>0.182</td>
<td>0.165</td>
<td>0.196</td>
<td>2</td>
<td>7</td>
<td>347</td>
<td>2.60</td>
<td>2.65</td>
<td>1.82</td>
<td></td>
</tr>
<tr>
<td>1982</td>
<td>0.010</td>
<td>0.031</td>
<td>0.014</td>
<td>0.057</td>
<td>2</td>
<td>7</td>
<td>221</td>
<td>1.95</td>
<td>1.95</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>1983</td>
<td>0.028</td>
<td>0.173</td>
<td>0.032</td>
<td>0.196</td>
<td>2</td>
<td>7</td>
<td>211</td>
<td>8.33</td>
<td>8.27</td>
<td>-0.67</td>
<td></td>
</tr>
</tbody>
</table>

(c) Bublitz, Frecka and McKeown [1985], table 5, p.19

<table>
<thead>
<tr>
<th>Year</th>
<th>Adj-R²_A</th>
<th>Adj-R²_B</th>
<th>R²_A</th>
<th>R²_B</th>
<th>df₁</th>
<th>df₂</th>
<th>obs.</th>
<th>Calcd. BFK's</th>
<th>%error</th>
<th>F-value</th>
<th>F-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>0.142</td>
<td>0.167</td>
<td>0.144</td>
<td>0.174</td>
<td>2</td>
<td>4</td>
<td>328</td>
<td>5.89</td>
<td>6.69</td>
<td>13.54</td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td>0.142</td>
<td>0.165</td>
<td>0.144</td>
<td>0.175</td>
<td>2</td>
<td>5</td>
<td>328</td>
<td>3.99</td>
<td>4.03</td>
<td>0.92</td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td>0.142</td>
<td>0.190</td>
<td>0.144</td>
<td>0.204</td>
<td>2</td>
<td>7</td>
<td>328</td>
<td>4.86</td>
<td>4.85</td>
<td>-0.28</td>
<td></td>
</tr>
<tr>
<td>1981</td>
<td>0.081</td>
<td>0.119</td>
<td>0.083</td>
<td>0.126</td>
<td>2</td>
<td>4</td>
<td>350</td>
<td>8.51</td>
<td>8.43</td>
<td>-0.88</td>
<td></td>
</tr>
</tbody>
</table>

+ apparent error in published table

(d) Murdoch [1986] pp.281-282

<table>
<thead>
<tr>
<th>Year</th>
<th>Adj-R²_A</th>
<th>Adj-R²_B</th>
<th>R²_A</th>
<th>R²_B</th>
<th>df₁</th>
<th>df₂</th>
<th>obs.</th>
<th>Calcd. Mur's</th>
<th>%error</th>
<th>F-value</th>
<th>F-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>0.07827</td>
<td>0.08740</td>
<td>0.078</td>
<td>0.093</td>
<td>1</td>
<td>2</td>
<td>160</td>
<td>2.59</td>
<td>2.57</td>
<td>-0.80</td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td>0.07827</td>
<td>0.08076</td>
<td>0.078</td>
<td>0.086</td>
<td>1</td>
<td>2</td>
<td>160</td>
<td>1.43</td>
<td>1.40</td>
<td>-2.15</td>
<td></td>
</tr>
<tr>
<td>1981</td>
<td>0.14342</td>
<td>0.14594</td>
<td>0.143</td>
<td>0.151</td>
<td>1</td>
<td>2</td>
<td>168</td>
<td>2.49</td>
<td>2.44</td>
<td>-3.53</td>
<td></td>
</tr>
<tr>
<td>1981</td>
<td>0.14342</td>
<td>0.15485</td>
<td>0.143</td>
<td>0.159</td>
<td>1</td>
<td>2</td>
<td>168</td>
<td>3.26</td>
<td>3.20</td>
<td>-1.80</td>
<td></td>
</tr>
<tr>
<td>1982</td>
<td>0.00123</td>
<td>0.00174</td>
<td>0.001</td>
<td>0.017</td>
<td>1</td>
<td>2</td>
<td>167</td>
<td>2.77</td>
<td>2.75</td>
<td>-0.56</td>
<td></td>
</tr>
<tr>
<td>1982</td>
<td>0.00123</td>
<td>0.00940</td>
<td>0.001</td>
<td>0.015</td>
<td>1</td>
<td>2</td>
<td>167</td>
<td>2.37</td>
<td>2.37</td>
<td>0.04</td>
<td></td>
</tr>
</tbody>
</table>
Appendix 3.2 continued. Calculation of F-statistics.

Panel B: Calculation of F-values for tests NOT reported in published papers

(a) Beaver and Landsman [1983] (adjusted R² not used by BL)

Model A regresses BL's RETURN variable against HC. Model B adds BL's POSTP variable to the regression.

<table>
<thead>
<tr>
<th>Year</th>
<th>Adj-R²_A</th>
<th>Adj-R²_B</th>
<th>R²_A</th>
<th>R²_B</th>
<th>df_A</th>
<th>df_B</th>
<th>obs.</th>
<th>Calculated F-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1979</td>
<td>0.22</td>
<td>0.24</td>
<td>1</td>
<td>2</td>
<td>392</td>
<td></td>
<td></td>
<td>10.26 **</td>
</tr>
<tr>
<td>1980</td>
<td>0.21</td>
<td>0.24</td>
<td>1</td>
<td>2</td>
<td>323</td>
<td></td>
<td></td>
<td>12.67 **</td>
</tr>
<tr>
<td>1981</td>
<td>0.08</td>
<td>0.09</td>
<td>1</td>
<td>2</td>
<td>297</td>
<td></td>
<td></td>
<td>3.24</td>
</tr>
</tbody>
</table>

**At 5%, F-value for sample of >120, 1 df in numerator is 3.9

(b) Bublitz, Frecka and McKeown [1985], Table 5.

Model A regressed abnormal return against 5 variables, two measuring change in HC income, two measuring change in realized holding gain, and one measuring unrealized holding gain in the year of test.

Model B added two additional variables, one for purchasing power gain or loss (PPGL) in the prior year, another for PPGL in the test year, together these measure change in purchasing power gain.

<table>
<thead>
<tr>
<th>Year</th>
<th>Adj-R²_A</th>
<th>Adj-R²_B</th>
<th>R²_A</th>
<th>R²_B</th>
<th>df_A</th>
<th>df_B</th>
<th>obs.</th>
<th>Calculated F-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>0.165</td>
<td>0.190</td>
<td>0.175</td>
<td>0.204</td>
<td>5</td>
<td>7</td>
<td>328</td>
<td>5.98 **</td>
</tr>
<tr>
<td>1981</td>
<td>0.128</td>
<td>0.143</td>
<td>0.137</td>
<td>0.157</td>
<td>5</td>
<td>7</td>
<td>350</td>
<td>4.02 **</td>
</tr>
<tr>
<td>1982</td>
<td>0.014</td>
<td>0.033</td>
<td>0.026</td>
<td>0.051</td>
<td>5</td>
<td>7</td>
<td>323</td>
<td>4.12 **</td>
</tr>
<tr>
<td>1983</td>
<td>0.098</td>
<td>0.097</td>
<td>0.108</td>
<td>0.112</td>
<td>5</td>
<td>7</td>
<td>361</td>
<td>0.80</td>
</tr>
</tbody>
</table>

**At 5%, F-value for sample of >120, 2 df in numerator is 3.07
Chapter 4. Computer-based Accounting Systems: Practice

The previous two chapters attempted to identify the characteristics of "good" accounting systems by reviewing the normative and empirical literature. This chapter begins the right-hand branch of the four-chapter Requirements Determination process illustrated in the lower half of Figure 1.1: it surveys the present state of computer-based accounting practice; Chapter 5 reviews computer-based accounting systems theory. Opportunities and problems identified in this chapter should indicate areas for improvements in future computer-based accounting systems.

Despite the widespread use of computer-based accounting systems (see survey evidence in Section 4.1 below) there has been very little research into the features that make one system or suite of systems better than another. It was therefore necessary to conduct some first-hand empirical studies specially for this thesis. It is unusual to report original empirical research in the "literature survey" stage of a thesis, but in this case there was no choice. Most of Section 4.1 reports on results of two mail surveys of general ledger system users. Mail surveys can provide useful, unambiguous answers to simple questions, like "Do you have a computer-based general ledger?", but to gain an understanding of how today's systems work, and what they are intended to do, it was necessary to examine some specific systems in depth. Sources of information were first-hand experience with PC software, visits to user sites, and vendor's documentation. It was mainly through first-hand experience and reading documentation that
some useful insights into the nature and purposes of computer-based accounting software began to emerge. These insights are discussed in Section 4.2.

4.1 An Overview of Computer-based Accounting Systems in Practice

Following Anthony [1965] and Keen and Scott-Morton [1978], computer-based accounting systems are usually classified as either transaction processing systems or decision support systems. Transaction processing accounting systems (TPAS) are used for capturing, processing, storing, and reporting, the sometimes millions of day-to-day transactions that occur in a firm each month. Decision support systems (DSS) use spreadsheets and financial modelling packages for budgeting, forecasting, and analysis purposes. DSS packages are not normally used for capturing and processing the large volumes of day-to-day transactions required for accountability reporting. Some firms still develop their own TPAS software but, as documented below, there is an increasing trend towards use of packages for both TPAS and DSS.

Sales revenues of software suppliers provide an indication of the economic importance of accounting TPAS and DSS software. Industry sources estimate combined 1988 sales of the mainframe TPAS suppliers MSA, McCormack and Dodge, and Software International\(^1\) were around $US600 million. PC-based TPAS systems and in-house developed systems add to this figure. In the area of DSS software, Lotus Development Corporation reported 1988 sales of $US469

\(^1\) There has been something of a "shake-out" in the TPAS industry. MSA has since been taken over by Dun and Bradstreet, which also owns McCormack and Dodge. Both are now part of Dun and Bradstreet Software. Software International has been taken over by Computer Associates.
million. It would therefore seem reasonable to suggest that 1988 world revenue for packaged accounting software was around $US2,000 million. This is only about 1% of world-wide hardware sales (in 1988 IBM and Digital's sales revenues were $US60,000 million and $US11,500 million respectively [Fortune, April 24, 1989, p.168]). Nonetheless, at $US2,000 million p.a., design, development, and marketing of accounting software is a significant economic activity.

In Chapter 2 (see Figure 2.3, p.2-9) it was decided that this thesis would concentrate only on routine accounting information systems, i.e., only on TPAS. The annual $US600 million market for mainframe TPAS is offered as evidence that even this more narrow focus is important enough to warrant academic study.

4.1.1 A description of TPAS software

This section provides a description of TPAS software gleaned from personal experience with PC-based software, software vendors' brochures (e.g., the MSA GL system brochure which is 46 pages long), text-books (e.g., Kraushaar and Sullivan [1986] contains a useful 30-page description of present-day computer-based TPAS systems), discussions with two of the big-six accounting firms, and a number of personal visits to companies using general ledger (GL) software. It provides an outline of GL system features that is tested in the following sections, Sections 4.1.2 and 4.1.3.

Computer-based TPAS are designed today as a series of modules that collect and process data for specific applications like accounts payable, sales order
entry, inventory, etc. These modules are often large systems in their own right. For instance, the Australian manager of British software supplier Quality Software Products Ltd estimates that his company's GL package consists of 200 programs containing about 700,000 lines of COBOL, including comments. User Manuals for each module are often 5 - 10 cm. thick (with pages printed both sides). Licence fees for mainframe modules range from $50,000 to $200,000, depending mainly on the size of the processor.

*** Figure 4.1 goes approximately here, see next page ***

The most common TPAS module is the General Ledger. Vendors find that to sell their General Ledger product it must be self-contained, yet able to interface with other software, either developed in-house, or from a competitor. It is therefore understandable that each different module has evolved with its own file structure, with its own record layouts, and with special-purpose programs written to update and report from these files. Figure 4.1 illustrates how these systems typically interact. In many firms, special pathways have then been written to funnel data from the heterogeneous sub-systems, which might be called "feeder systems", into the GL. The GL is usually updated in batch mode.

In many respects computer-based GL systems are no different to the conventional T-account model used in the classroom. They are still required to accept journal entries (including standing or pro-forma journal entries), calculate account balances, and print "trial" balances. For database design reasons, account balances in computer GL systems are usually stored in one part of the database (often called an account file) and transaction data in another
Specialized sub-systems

- Sales Order Entry
- Sales Invoicing
- Accounts Receivable
- Inventory
- Product Costing
- Fixed Assets
- Material Requirements Planning
- Purchasing
- Accounts Payable
- Personnel
- Payroll
- Share Register

Summary journal entries → Computer-based General Ledger

Direct journal entries

Figure 4.1: Structure of a possible computer-based TPAS
(the journal entry file), but from a logical level the data are no different to T-accounts.

The big difference between computer-based ledger systems and the classroom model is that computer-based systems have powerful reporting and inquiry capabilities. These offer (i) comparisons with the past and with budget, (ii) flexibility, both to change the structure of existing reports and to generate ad hoc reports as required, and (iii) powerful transactions search and retrieval (audit trail) features. To provide data for comparisons with the past and with budget the computer GL database must be much more complex than the classroom model; it must store past account balances and past and future budget information.

*** Figure 4.2 goes approximately here, see next page ***

A representative large-system computer GL account record, reprinted (with permission) from a McCormack and Dodge training manual, is shown in Figure 4.2. This record is variable length, potentially up to 1885 bytes long per account. Most of this space is used for an array of up to fourteen by thirteen "monthly" numeric fields of current and historical information. The fourteen possible segments, shown at the right-hand side of Figure 4.2, are used to store monthly figures for each GL account as follows:

- Four years of actual figures (current year and prior three years)

---

2 Some firms operate with 13 four-week months per year.
# Chart of Accounts Design

## Chart of Accounts Design

<table>
<thead>
<tr>
<th>ACCOUNT TYPE</th>
<th>ACCOUNT DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BALANCE LIMITS</td>
<td>TRANSACTION LIMITS (in CR)</td>
</tr>
<tr>
<td>JOURNAL VALIDATION</td>
<td>SUB-LEDGER VALIDATION</td>
</tr>
<tr>
<td>MISCELLANEOUS IDENTIFICATION FIELDS</td>
<td></td>
</tr>
<tr>
<td>MISC A</td>
<td>MISC B</td>
</tr>
<tr>
<td>FLEXIBLE BUDGETING</td>
<td></td>
</tr>
<tr>
<td>STATEMENT LINE ADDRESS (by Book)</td>
<td></td>
</tr>
<tr>
<td>VARIANCE %</td>
<td>VARIANCE AMOUNT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACCOUNT</th>
<th>CENTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORP</td>
<td>ACCOUNT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CURRENT YEAR ACTUAL (by period)</th>
<th>CURRENT YEAR AVERAGE (by period)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAST YEAR ACTUAL (by period)</td>
<td>LAST YEAR AVERAGE (by period)</td>
</tr>
<tr>
<td>SECOND YEAR ACTUAL (by period)</td>
<td>SECOND YEAR AVERAGE (by period)</td>
</tr>
<tr>
<td>BUDGET 1 (by period)</td>
<td>BUDGET 1 (by period)</td>
</tr>
<tr>
<td>BUDGET 2 (by period)</td>
<td>BUDGET 2 (by period)</td>
</tr>
<tr>
<td>BUDGET 3 (by period)</td>
<td>BUDGET 3 (by period)</td>
</tr>
<tr>
<td>BUDGET 4 (by period)</td>
<td>BUDGET 4 (by period)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MISCELLANEOUS AMOUNT BALANCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCH ACTIVITY</td>
</tr>
</tbody>
</table>

156 characters
Six years of budget figures (e.g., for first, revised, and planned budgets)

Four years of average balances (current year and prior three years)

There is no place for any of this historical or budget information in the classroom GL model, yet it dominates the practical computer GL system database. In addition, many GL systems allow users to store non-monetary data. For instance, a firm may store details of sales of product X in physical units, by month, in its GL. Again, these figures have no place in the classroom GL model but they provide useful comparative and trend information to management.

In addition to offering comparisons with the past and with budget, computer GL systems also give users the ability to select and aggregate data (including transaction data) in many different ways. This capability is useful for two reasons. First, organization structures tend to change over time, so new or different reports are often required. Second, the scale of computer GL systems is so large (16% of respondents reported having more than 10,000 accounts) that selective reporting is desirable. In both cases the GL system acts like a special-purpose database query language for selecting and aggregating data.

The discussion above identified reporting and inquiry as the major reason for difference between computer-based GL systems and the classroom T-account model. But there are also other significant differences. Indeed, one of the most striking aspects of the large TPAS software packages is the huge range of special features built into them. Other features offered by many computer-based GL systems, that are not found in the classroom GL model, include:
• powerful security systems to restrict who can access and update information
• comprehensive user-defined editing of input data
• comprehensive audit trail information
• links to the outside world, for data from feeder systems, or for downloading and uploading data to and from PCs (including real-time update of mainframe files using data received from PCs)
• variable budgeting systems, including curve fitting from historical data, long-range planning, "what-if" analysis
• sophisticated automatic cost allocation procedures
• foreign currency translation
• interactively user-defined display screens
• multi-company consolidations

Different combinations and emphases in this vast array of ease-of-use features seems to be what distinguishes one large GL system from another. Note, however, that the basic accounting model of debits and credits of money amounts to objects called accounts is unquestioned, and is, unquestionably, the same across all systems.

4.1.2 Two mail surveys of TPAS software

In the past few years the author has participated in two mail surveys of Australian firms using TPAS software. The first, in 1988, concerned TPAS generally. The second, in 1989, concentrated on general ledgers. Both provide evidence that confirm the above description of TPAS software.
For the 1988 study [Seddon and Wong, 1989] a seven-page questionnaire, based mainly on features listed in various software vendors' advertising materials, was developed and tested with three face-to-face interviews. A random sample of 198 manufacturing firms, all with 200 or more employees, was then drawn from the 1984 Victorian Manufacturing Directory prepared by the Department of Industry, State of Victoria. Letters were mailed to "the senior accounting officer" of each firm in June, 1988, with a request that the accompanying questionnaire should be directed to "the accountant who is familiar with the computer-based general ledger in your organization". The first-round response rate was 50%, and this increased to 63% after a second-round follow-up letter. Chi-square tests on each of 33 questions indicated no significant differences between first- and second-round respondents. Thus the responses from the questionnaire may be treated as representative of large Australian manufacturing firms generally.

For the 1989 study [Seddon and Yip, 1990] another seven-page questionnaire was again prepared, this time using a combination of questions from Ives et al. [1983], Baroudi and Orlikowski [1988], Doll and Torkzadeh [1988], and the 1988 questionnaire. A random sample of 200 firms with 200 or more employees was drawn from Business Who's Who of Australia, 1988 edition (firms from the 1988 questionnaire were excluded). The first-round response rate was 41%, and this increased to 50% after a second-round follow-up. Goodman-Kruskal gamma statistics [Siegel and Castellan, 1988, pp.291-8] of each of the 62 questions indicated that later respondents scored their GL system significantly lower than early respondents. Assuming that non-respondents were more like later
respondents, the 1989 results probably apply to a more-satisfied group of users than average.

Descriptive data from the two surveys are summarized in Table 4.1. They show that for larger Australian companies (200 or more employees) computer-based GL systems are in almost universal use. Most users now use packaged GL software, and most users are quite satisfied with their systems.

<table>
<thead>
<tr>
<th></th>
<th>1988</th>
<th>1989</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of respondents (of 200 questionnaires mailed out)</td>
<td>63%</td>
<td>50%</td>
</tr>
<tr>
<td>Number of respondents who were financial accountants</td>
<td>73%</td>
<td>85%</td>
</tr>
<tr>
<td>Number who were using a computer general ledger (GL)</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td>Number who were using packaged GL software (cf. custom)</td>
<td>75%</td>
<td>70%</td>
</tr>
<tr>
<td>Number describing themselves as &quot;satisfied,&quot; or better</td>
<td>89%</td>
<td>93%</td>
</tr>
<tr>
<td>Number whose firms had replaced their first computer GL with a newer, presumably better, one</td>
<td>55%</td>
<td>66%</td>
</tr>
<tr>
<td>Ratio of packages to custom-built GL systems installed since 1983</td>
<td>87%</td>
<td>not asked</td>
</tr>
</tbody>
</table>

Table 4.1: Descriptive data from the two surveys

4.1.2(a) Findings from the 1988 study

Figure 4.1 (see previous section) was included in the 1988 questionnaire, and respondents were asked to tick (a) modules used in their firm, and (b) modules directly coupled, or integrated, with the GL (so that an entry in a feeder system led to the GL being updated automatically). Responses are given in Table 4.2. They show that after the GL system, Accounts Payable is by far the most
popular module (93% of respondents with a package GL also had an AP module),
and almost all of those systems (88%) were integrated with the GL. Other
"popular" packaged software modules, in descending frequency of use, were
Accounts Receivable (81%), Sales Invoicing (68%), Inventory (63%), Sales Order
entry (49%), Fixed Assets (49%), Product Costing (48%), and Payroll (47%).

Columns 4 and 5 in Table 4.2 show that most Accounts Payable systems (and
many other TPAS) are integrated with the GL system so that transactions do not
have to be re-keyed.

*** Tables 4.2 and 4.3 go approximately here, see next page ***

Relationships between various database features and Overall Satisfaction are
given in Table 4.3. Column 2 shows the overall percentage of the 125
respondents who answered "Yes" to having and using various features of their
GL systems. If it is assumed that some sort of economic Darwinism [Watts and
Zimmerman, 1986, p.195] leads packages with useful features to survive in the
competitive marketplace, and packages with less useful features to fail, the
figures in column 2 give some idea of which features are more important than
others. For instance, economic Darwinism suggests that if 83% of systems store
cost allocation data, but only 54% of systems store statistical data, then the
former feature is more useful than the latter. In the case of the top four lines
of Table 4.3, the two alternatives must be added before applying the economic
Darwinism test. In both cases the result is 100%, which indicates that the
ability to store history and budget information, usually summarized by month, is
very important to GL system users.
Table 4.2: Modules used in addition to Computer-based General Ledger (1988 study)

<table>
<thead>
<tr>
<th>column number (1)</th>
<th>% of firms with module</th>
<th>Number integrated as a % of those having</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Package GL (2)</td>
<td>Custom-built GL (3)</td>
</tr>
<tr>
<td>Accounts Payable</td>
<td>93</td>
<td>88</td>
</tr>
<tr>
<td>Accounts Receivable</td>
<td>81</td>
<td>70</td>
</tr>
<tr>
<td>Sales Invoicing</td>
<td>68</td>
<td>73</td>
</tr>
<tr>
<td>Inventory</td>
<td>63</td>
<td>64</td>
</tr>
<tr>
<td>Sales Order Entry</td>
<td>49</td>
<td>55</td>
</tr>
<tr>
<td>Fixed Assets</td>
<td>49</td>
<td>50</td>
</tr>
<tr>
<td>Product Costing</td>
<td>48</td>
<td>47</td>
</tr>
<tr>
<td>Payroll</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>Purchasing</td>
<td>41</td>
<td>52</td>
</tr>
<tr>
<td>Material Req.Planning</td>
<td>26</td>
<td>20</td>
</tr>
<tr>
<td>Personnel</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Number of firms (denominator for %)</td>
<td>91</td>
<td>34</td>
</tr>
</tbody>
</table>

- = percentage not meaningful since denominator less than 20
Table 4.3: Relationships between content of GL system database and Overall Satisfaction (1988 study)

<table>
<thead>
<tr>
<th>Database feature</th>
<th>Overall % of people who answered &quot;Yes&quot; to having feature</th>
<th>Percentages classified by Overall Satisfaction Code</th>
<th>Gamma Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(2)</td>
<td>(3)</td>
<td>(4)</td>
</tr>
<tr>
<td>Column number (1)</td>
<td>(1)</td>
<td>(2)</td>
<td></td>
</tr>
<tr>
<td>Current year history only</td>
<td>63</td>
<td>90</td>
<td>38</td>
</tr>
<tr>
<td>OR Multiple years history</td>
<td>37</td>
<td>10</td>
<td>17</td>
</tr>
<tr>
<td>Current year budget only</td>
<td>64</td>
<td>91</td>
<td>92</td>
</tr>
<tr>
<td>OR Multiple years budget</td>
<td>36</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Statistical (non-monetary) data</td>
<td>54</td>
<td>46</td>
<td>36</td>
</tr>
<tr>
<td>Account and reporting relationships</td>
<td>90</td>
<td>54</td>
<td>88</td>
</tr>
<tr>
<td>Report specifications</td>
<td>89</td>
<td>69</td>
<td>80</td>
</tr>
<tr>
<td>Cost allocation data</td>
<td>83</td>
<td>77</td>
<td>80</td>
</tr>
<tr>
<td>Foreign currency translation</td>
<td>34</td>
<td>28</td>
<td>38</td>
</tr>
<tr>
<td>Overall % who answered &quot;Yes&quot;</td>
<td>125</td>
<td>13</td>
<td>25</td>
</tr>
</tbody>
</table>

** probability that this statistic could occur by chance < 1%
* probability < 5%
+ probability < 10%
$ not significantly related to satisfaction, but as over 90% of second-system-and-later respondents had this feature it must still be important.

For instance, data line 6, column 3, shows that 54% (7 of 13) of Dissatisfied users (users coded 0 or 1 in response to the Overall Satisfaction question) said their GL systems could store Account and Reporting relationships in their database.
Respondents to the 1988 study were asked to rate their Overall Satisfaction with their GL system on a scale from zero to four as follows: Very dissatisfied = 0, Dissatisfied = 1, Neutral = 2; Satisfied = 3, and Very Satisfied = 4. Grouping 0 and 1 responses together, columns 3 - 6 of Table 4.3 analyze the figures in column 2 by Overall Satisfaction. The pattern evident in columns 3 - 6, of more satisfied users having more stored data, is tested by the gamma statistics in column 7. The highly "significant" bivariate test statistics\(^3\) in column 7, lines 1, 2, 4, and 5 suggest that GL systems that store multiple years of history and budget information in the database, and/or account relationships and report specifications in the database, are more likely to have satisfied users.

An unexpected finding was that when first-GL-system users were separated from second-system-and-later users, "significance" tests for the two groups were often different; gamma statistics for the two groups are shown in columns 8 and 9 of Table 4.3. In the main, "significant" relationships in column 7 are due to either first or second system users, but rarely due to both. In many cases almost all second-system-and-later users stored the data so there was no variance to be correlated with satisfaction. For instance, the relationship between stored Report Specifications and Overall Satisfaction is not "significant" for second-system-and-later users, but as 93% of them indicated

\(^3\) It is not claimed that these "significance" tests indicate the true significance of each of the variables in explaining Overall Satisfaction; they are merely indicative of the underlying relationships in the data. For these tests to be true measures of significance it would be necessary to control for all other potentially relevant variables, which is clearly not the case since more than one variable in Table 4.3 is flagged as "significant". A more rigorous multiple regression analysis of the 1989 data is provided in Section 4.1.2(b). What Tables 4.3 and 4.4 do show is that there are systematic differences between presence or absence of certain GL system features and Overall Satisfaction.
they had this feature (indicated by the $'s$ beside column 9) the absence of a "significant" relationship does not mean stored report specifications are unimportant.

*** Table 4.4 goes approximately here, see next page ***

In addition to the above database features, data were also collected about 17 specific features of GL systems. Table 4.4 summarizes these responses. As in Table 4.3, column 2 shows the overall percentage of users who answered "Yes" to having and using various features of their GL systems. By the economic Darwinism argument, these percentages give some idea of which features are more important than others. Columns 3 through 6 of Table 4.4 classify column 2 figures by Overall Satisfaction and express the results as percentages of each satisfaction category. Column 7 tests, on a line-by-line basis, the "significance" of relationships between use of a given feature and Overall Satisfaction. Columns 8 and 9 show that users of second-system-and-later GL systems again exhibit very different patterns of interests to first-system users.

Finally, a regression analysis with the 1988 data suggested that users' Overall Satisfaction with their GL systems was principally determined by the system's reporting features. Second-system users, who presumably had good reporting, were interested in on-line inquiry and possibly budgeting. This invited questions about the meaning of "reporting usefulness". Did it mean that the information in the reports was useful, or were users keen on the ease with which reports could be generated? This question was answered in the 1989 study.
Table 4.4: Relationship between use of GL system features and Overall Satisfaction (1988 data)

<table>
<thead>
<tr>
<th>General Ledger system feature</th>
<th>Overall % of people who answered &quot;Yes&quot; to having feature</th>
<th>Percentages classified by Overall Satisfaction Code</th>
<th>Gamma Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1) (2) (3) (4) (5) (6) (7) (8) (9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Column number</strong> (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data Capture</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-line data entry</td>
<td>86</td>
<td>85 68 90 93</td>
<td>0.39 * 0.05 0.74 **</td>
</tr>
<tr>
<td>Standing journal entries</td>
<td>78</td>
<td>38 84 83 82</td>
<td>0.31 * 0.32 0.24 $</td>
</tr>
<tr>
<td>Auto-reformat feeder jes</td>
<td>66</td>
<td>23 68 73 68</td>
<td>0.29 * 0.32 0.17</td>
</tr>
<tr>
<td>Automatic reversal jes</td>
<td>74</td>
<td>54 76 78 71</td>
<td>0.13 * 0.32 -0.20</td>
</tr>
<tr>
<td>Real-time update</td>
<td>57</td>
<td>46 32 66 64</td>
<td>0.30 * -0.01 0.59 **</td>
</tr>
<tr>
<td><strong>Inquiry</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-line inquiry</td>
<td>92</td>
<td>83 85 97 96</td>
<td>0.63 ** 0.48 0.32 *</td>
</tr>
<tr>
<td>Borderless inquiry</td>
<td>23</td>
<td>0 16 29 29</td>
<td>0.42 ** 0.44 0.40 *</td>
</tr>
<tr>
<td><strong>Reporting</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-dimensional reports</td>
<td>87</td>
<td>62 80 93 93</td>
<td>0.61 ** 0.76 0.32 $</td>
</tr>
<tr>
<td>Custom reports</td>
<td>72</td>
<td>31 60 83 79</td>
<td>0.45 ** 0.56 0.11 $</td>
</tr>
<tr>
<td>Consolidations</td>
<td>65</td>
<td>31 52 80 61</td>
<td>0.29 * 0.19 0.35 +</td>
</tr>
<tr>
<td>Consolidate diff. chart of accounts</td>
<td>47</td>
<td>23 44 49 57</td>
<td>0.27 * 0.14 0.34 *</td>
</tr>
<tr>
<td>Consolidate diff. accounting periods</td>
<td>10</td>
<td>0 0 7 29</td>
<td>0.84 ** (4 obs) 0.75 **</td>
</tr>
<tr>
<td><strong>Budgeting</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use multiple budgets</td>
<td>42</td>
<td>15 20 49 61</td>
<td>0.46 ** 0.53 0.33 *</td>
</tr>
<tr>
<td>Use budget assistance programs</td>
<td>43</td>
<td>23 36 44 57</td>
<td>0.29 * 0.44 0.04 *</td>
</tr>
<tr>
<td>PC download/upload</td>
<td>42</td>
<td>31 48 41 46</td>
<td>0.10 0.22 0.03</td>
</tr>
<tr>
<td><strong>Cost allocation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simple cost allocation</td>
<td>78</td>
<td>62 64 85 86</td>
<td>0.41 ** 0.47 0.36 +</td>
</tr>
<tr>
<td>Complex cost alloc. (simultaneous eqn.)</td>
<td>17</td>
<td>38 12 15 14</td>
<td>-0.19 -0.25 -0.11</td>
</tr>
<tr>
<td><strong>Number of respondents</strong></td>
<td>125</td>
<td>13 25 59 28</td>
<td>125 53 68</td>
</tr>
</tbody>
</table>

** probability that this statistic could occur by chance < 1%
* probability < 5%
+ probability < 10%
$ not significantly related to satisfaction, but as over 80% of second-system-and-later respondents had this feature it must still be important.

For instance, on the first line of the table, column 3 shows that 85% (11 of 13) of Dissatisfied users (users coded 0 or 1 in response to the Overall Satisfaction question) said they used On-line data entry in their GL system.

Multidimensional reporting was defined in the questionnaire to mean "report by division, department, product-line, region, or any other breakdown".
4.1.2(b) Findings from the 1989 study

Motivated by this question from the 1988 study, the 1989 study set out to investigate the nature of reporting usefulness. Compared to the tentative explorations of the 1988 study the 1989 study was much more rigorous. Its satisfaction measures were based on the literature on measuring User Information Satisfaction (UIS), particularly Ives, Olson, and Baroudi [1983] and Doll and Torkzadeh [1988]. It also controlled for variables other than Reporting Usefulness that had been reported in the literature as influencing UIS.

Ives et al.'s factor analysis [1983] suggested that three variables determined user information satisfaction: Relationship with the EDP department (EDP), Knowledge and involvement of the user in system development (KI), and the information content of reports, termed the Information Product (IP). However, Doll and Torkzadeh's factor analysis [1988] suggested that two variables determined end-user satisfaction: Information Product and Ease of Use (EASE). For the 1989 study these four variables were measured in addition to measures of the usefulness of various features of GL systems (FEATURES) such as report generating, online inquiry, budgeting, and so on. The final regression from the 1989 study [Seddon and Yip, 1990, Table 8, Regression 8.4] was as follows:

<table>
<thead>
<tr>
<th>const.</th>
<th>EDP</th>
<th>EDP_M</th>
<th>KI</th>
<th>IP</th>
<th>FEATURES</th>
<th>n</th>
<th>Adj. R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.75</td>
<td>-0.10</td>
<td>0.14</td>
<td>0.13</td>
<td>0.85</td>
<td>0.44</td>
<td>88</td>
<td>0.73</td>
</tr>
<tr>
<td>(t)</td>
<td>(-1.23)</td>
<td>(1.08)</td>
<td>(1.73)</td>
<td>(7.48)</td>
<td>(4.45)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>+</td>
<td></td>
<td>***</td>
<td>***</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*** p<0.001,  + p<0.1
In this regression, the dependent variable Overall Satisfaction was measured quite reliably with four scales (Cronbach alpha 0.954), and using Cohen and Cohen's advice [1983] a dummy variable EDP Missing (EDP_M) was included for the 35 cases where respondents did not have an EDP department. The regression results showed that the information content of the reports (IP) was the principal determinant of overall satisfaction, not the ease-of-use of the GL system's reporting features. The second significant variable was the average of the scores for satisfaction with four or more GL system features (FEATURES). However, none of the features variables by itself, e.g., reporting, online inquiry, or budgeting, was significant.

4.1.3 Summary, mail surveys

These two mail surveys collected information about both general ledger systems and feeder systems. Conclusions are as follows:

(i) Computer general ledger systems are very widely used. In both studies, 99% of firms with 200 or more employees reported using computer-based GL systems. Most companies installing a GL system today buy packaged software, rather than have it custom-built; many have installed their second computer-based GL system.

(ii) As shown in Figure 4.1, much of the data in those general ledgers are first captured in various feeder systems, then funnelled through to the general ledger. These feeder systems contain much richer descriptions of
their application areas than traditional accounting systems of, say, thirty years ago.

(iii) Computer-based GL systems typically consist of a number of programs for data capture, inquiry, reporting, budgeting, cost allocation, and chart of accounts maintenance. Taken together, these features also influence overall satisfaction, but it is not possible to identify any one feature that "makes or breaks" a GL package.

(iv) The key determinant of satisfaction of GL system users is the quality of the information product, i.e., the content, accuracy, format, and timeliness of reports.

4.2 A Detailed Review of Five Computer-based TPAS

Where the previous section provides an overview of computer-based accounting systems, this section examines two large-computer general ledger packages, two PC-based integrated accounting systems, and one large-computer fixed-asset system in more depth. The packages reviewed in this section are listed in Table 4.5. As Weber [1986] has observed, computer software packages may be used by many hundreds, perhaps thousands, of organizations around the world, so a review of a representative range of packages is a simple way to gain insights into the accounting requirements of many organizations.
Table 4.5: Packages Reviewed in Section 4.2

<table>
<thead>
<tr>
<th>Package</th>
<th>Target computer</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>McCormack and Dodge's Millennium General Ledger System, GL:M</td>
<td>IBM mainframe</td>
<td>Large-computer GL system</td>
</tr>
<tr>
<td>Praxa Ltd's Starfire-II General Ledger System</td>
<td>DEC VAX</td>
<td>Representative mid-size GL system</td>
</tr>
<tr>
<td>Attache 4</td>
<td>IBM PC and clones</td>
<td>Very simple all-in-one system designed to record general ledger, accounts receivable, accounts payable, and inventory for the small business</td>
</tr>
<tr>
<td>Computer Associate's ACCPAC Plus</td>
<td>IBM PC and clones</td>
<td>Comparable in power to the Starfire system for the VAX</td>
</tr>
<tr>
<td>MSA's Fixed Asset System&lt;sup&gt;4&lt;/sup&gt;</td>
<td>IBM mainframe</td>
<td>Large-computer fixed-asset system</td>
</tr>
</tbody>
</table>

4.2.1 The McCormack and Dodge Millennium System, GL:M

The large general ledger accounting system investigated for this study is GL:M Version 2.02, from McCormack and Dodge (M&D). M&D is a subsidiary of the large US corporation, Dun and Bradstreet. GL:M runs on IBM and Fujitsu mainframes, and since 1988, on VAX systems as well. The entire system is written in COBOL. Licence fees vary from around $60,000 to $180,000 depending on system size, with annual maintenance and support charges thereafter. Melbourne-based customers include some of Australia's largest companies, e.g., Coles Myer, CRA, ICI, Toyota, State Bank of Victoria, and National Mutual Royal Bank. At a recent M&D user group meeting in Melbourne, the largest company, Coles Myer, reported having 700,000 general

---

<sup>4</sup> In 1990, MSA was taken over by McCormack and Dodge's parent company, Dun and Bradstreet.
ledger accounts. They process over 12,000,000 purchase transactions through their M&D accounts payable system (interfaced to the general ledger) each year. A smaller company, ICI Australia, had 94,000 accounts containing records for 130 companies.

GL:M is a product that has evolved over many years. It now uses IBM VSAM (indexed sequential) files, and may be updated directly, one transaction at a time if need be. Batch processing of large groups of transactions is widely used because it is more efficient, and data tend to come from feeder systems in batches anyway. At the heart of the system are three files:

- Master Data Base
- Current Transaction Database
- Control file.

The Master Data Base stores general ledger account records. The Transaction Database stores journal entry lines. The Control file stores all the rules for account codes, reporting hierarchies, accounting periods, predefined journal entry templates, user specified reports, security, and so on. A site visit to the National Mutual Royal Bank (NMR Bank) revealed that it had 45,000 general ledger accounts, and 260,000 journal entry lines posted per month. To reduce processing overhead, transactions on the Current Transaction Database are archived, on a monthly basis, to a tape file called the Historical Transaction Database, but conceptually the database is unchanged by this archiving process.
Details of the account record for the Master Data Base were given in Figure 4.2 in Section 4.1.1 above. Record lengths are variable up to 1885 bytes long, but a sales representative from M&D said that in his experience account records averaged about 600 bytes long. In the NMR Bank system 85MB of disk space was required for this one file. Most of the space in each record is used for the array of up to fourteen by thirteen "month" numeric fields (COBOL PIC S9(13)V99, stored in packed decimal format in 8 bytes) of current and historical, actual and budget, information.

General ledger accounts are of six types:

<table>
<thead>
<tr>
<th>Account type</th>
<th>Code</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset account</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Liability account</td>
<td>L</td>
<td>this class includes Owners' Equity</td>
</tr>
<tr>
<td>Income account</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>Expense account</td>
<td>E</td>
<td>e.g., to record sales units, or hours worked, each month</td>
</tr>
<tr>
<td>Volume account</td>
<td>V</td>
<td>e.g., to record inventory level of a particular product at month end</td>
</tr>
<tr>
<td>Balance account</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

The V- and B-type accounts do not contain money account balances, and are excluded from normal general ledger reports. They exist as a convenient means of classifying non-monetary information and as the source of data for flexible budgets. For instance, if the budget for certain manufacturing expense accounts is thought to be related to units produced, or hours worked, this

5 Such arrays for each account are quite normal. Another large system studied, APS/GL, marketed in Australia by Price Waterhouse for VAX and Pick systems, had maximum array size of 99 segments, each with 16 "months" for each account.
statistical information may be captured each month, stored in a V-type account, and used for calculating the corresponding budget expenses.

At the end of each year the figures for the current 13 months are rolled-back to the previous year, balances for A, L, and B accounts are brought forward to the first month of the new year, and balances for I, E, and V accounts are set to zero for the new year.

The identifier for each account record is a 23 byte (maximum) alpha-numeric account identifier consisting of three parts:

<table>
<thead>
<tr>
<th>Corporation code</th>
<th>3 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account number</td>
<td>10 bytes</td>
</tr>
<tr>
<td>Centre</td>
<td>10 bytes</td>
</tr>
</tbody>
</table>

Not all 23 bytes have to be used. For example, five accounts illustrated in the McCormack and Dodge training manual are only 14 bytes long, as follows:

<table>
<thead>
<tr>
<th>CC</th>
<th>A/C</th>
<th>Centre</th>
<th>Account Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>4000</td>
<td>DUANE-BO</td>
<td>Sales of product Duane in Boston</td>
</tr>
<tr>
<td>01</td>
<td>4000</td>
<td>DUANE-CH</td>
<td>Sales of product Duane in Chicago</td>
</tr>
<tr>
<td>01</td>
<td>4000</td>
<td>DUANE-LA</td>
<td>Sales of product Duane in Los Angeles</td>
</tr>
<tr>
<td>01</td>
<td>4000</td>
<td>JOLLA-BO</td>
<td>Sales of product Jolla in Boston</td>
</tr>
<tr>
<td>01</td>
<td>4000</td>
<td>JOLLA-CH</td>
<td>Sales of product Jolla in Chicago</td>
</tr>
</tbody>
</table>

The above example shows how a fixed-structure account-identifier based on Corporation, Account, and Centre, can make it difficult to define reports. A four-part structure that enabled sales to be GROUPED BY CITY (Boston, Chicago, Los Angeles, etc.) or by PRODUCT (Duane, Jolla, etc.) would have
made report definition easier. Ideally the system should be like a relational database, with no restriction on the number of record selection criteria and GROUP BY fields. In GL:M, M&D bypass this problem by allowing the user to define up to 30 aggregation hierarchies that enable the system to summarize automatically all sales from Boston, or all sales of product Duane. According to the M&D representative the newly released GL product, Version 3, has no such restrictions on the structure of the account code.

Records in the Transaction files in the M&D system are exactly 200 bytes long. Thus in the NMR Bank, for instance, two months of journal entries occupied another 50MB of disk space. These records are all directly accessible for reporting and audit trail purposes. Furthermore, they were quite heavily used for reporting purposes. Journal entries may be entered as manual, automated, repeating, skeletal, or translated journal transactions. For manual entries, the user keys-in account identifiers and money values; automated entries simplify automatic reversal of end of period accruals; repeating entries are the same month after month; skeletal entries use the same accounts each month, but the user keys-in the money values; translated entries are generated by feeder systems like accounts payable and fixed assets. The majority of the 260,000 monthly entries in the NMR Bank’s system came from feeder systems.

There is considerable emphasis on budgeting, reporting against budget, and flexible budgeting in the M&D GL system. In the last few years it has been

---

6 "GROUP BY" is an SQL Command. SQL is a relational database query language developed initially by IBM in the 1970s, and now available on the majority of database systems.

4 - 20
possible to download, and upload information to and from a PC. For instance, it is now possible to upload a comprehensive budget prepared using a spreadsheet program on a PC (each record is structured as if it is feeding information into successive fields of the on-line update screen). One may speculate that in future much of the curve fitting and long-range planning will be better performed using a spreadsheet on a PC. The M&D representative confirmed that because PCs are now so widely used, some of GL:M Version 2's budgeting features have been dropped in Version 3.

Why would an organization buy such a system? Why would an organization want to maintain an array of $45,000 \times 6 \times 13 = 3,510,000$ facts about itself (Coles Myer have $700,000 \times 8 \times 13 = 72,800,000$ facts!) in its general ledger? When this question was posed to the M&D customer support representative, he replied

"We only give the customers what they want. The question that puzzles us is why the users want to keep their transaction histories on-line for so long. Some of them keep a whole year's history, millions of records, on-line."

Remember that these are all facts related to the general ledger. The feeder systems like accounts payable, fixed assets, and industry specific systems, also store many additional details.

The two questions in the previous paragraph may be answered in two stages. First, it is clear that all these megabytes of information are intended for internal, not external, consumption. The users are management, not investors or creditors. Second, management seems to want a detailed time-series of
changes in its financial position for planning and evaluation purposes. Evidently, management wants at its fingertips very detailed information like the sales of product Duane in Boston each month for the last three to four years, together with corresponding budget figures. When the senior accountant of the NMR Bank (45,000 general ledger accounts) was asked why all these facts were needed, he answered, "How else would you expect us to run our business". Third, the desire to maintain long transaction histories may be explained as either (i) accounting staff cross-checking past transactions to ensure that they were recorded correctly, or (ii) search activity arising because the classifications in the general ledger were not specific enough to be useful directly. Discussions at the 1988 M&D Melbourne User Group meeting suggest explanation (ii), searching for details of specific transactions, is the more common reason.

4.2.2 The Praxa Ltd Starfire-II System

Praxa is a Melbourne-based company specializing in software for DEC VAX computers. Their medium-sized Starfire system is written in COBOL and is based on programs developed by Mini-Computer Business Applications, Inc., of Glendale, California. The system is described as "Financial control, Distribution, and Manufacturing software for Australian conditions". The Starfire general ledger system is like a small-scale, less flexible, version of the M&D system (Section 4.2.1).

Account numbers in Starfire may have a user-defined format of up to 21 digits (only numbers allowed) consisting of so-called "locational" and "descriptive" parts. Locational information in the account number is used to define financial
entities, like companies and departments. Descriptive information in the account number is used to define accounts within entities. The descriptive part of the account number is used to define ranges of accounts to appear in financial reports.

As with the larger systems, each account in the general ledger contains standard information like Description, Counter-balance account, Automatic distribution code, trial balance subtotal level, financial statement type (Balance sheet or Profit and Loss), parenthesis control code (to control brackets around credits in accounts that typically have debit balances, and vice versa). As with the larger systems each account also stores up to 13 periods of budget-type information, but there is only provision for two segments (one budget, and one "comparison" segment). By comparison, M&D had 14 segments, and MAPS/GL (a package marketed by Price Waterhouse Urwick) had 100. Unlike the big systems, Starfire uses a technique called "compressing" the transaction file for the history of actual journal entries. All journal entries in a specified period can be "compressed" by date, period, source within date, or source within period. For instance, if journal entries were compressed by date, all journal entries to one account on one day would be deleted and replaced by a single net change journal entry. Compression is irreversible.

Journal entries in Starfire are very simple records; they may be entered from many terminals simultaneously. For manual journal entries the user specifies the date, account and amount to be debited or credited, a cross reference, whether automatic reversal is to take place next period, or automatic distribution this period. Journal entries to dates outside the current period are
flagged on the edit listing. As a user is entering the journal entry, he/she has the option of pressing the "gold" key for automatic generation of a counter-balancing journal entry. Recurring journal entries contain fields identical to manual entries. They may be either permanent (same money amount each period) or variable (user recalls the entry and changes the amount). Journal entries are maintained on a transaction file, and the user may inquire, change, and/or delete any journal entry up to the time it is posted to the ledger. The transactions file may be searched sequentially, or by account number, or by transaction date (and time).

Automatic journal entries may be generated from the accounts receivable, accounts payable, fixed assets, payroll, job costing, and inventory subsystems. These are collected via a program called the General Ledger Interface application. The manual describing the Interface program is full of warnings about minimal data validation taking place, the need for backup copies of files, problems if the transactions are posted to the wrong company, or wrong period, and mismatches with earlier releases of the various packages. User beware!

Unlike the larger systems, many of the reports in Starfire, like trial balance, edit listing, ledger worksheet, and ledger account inquiry, are predefined. However, Starfire also assumes the user will want to define ranges of accounts for reports of type Profit and Loss, Balance Sheet, and Cash Flow. For the latter reports, layouts are pre-specified, with fixed columns, but the user specifies ranges of accounts, whether they are to be printed or accumulated, and when subtotals are to be printed and cleared. Here again, there is plenty of scope for user error. The system even provides a program to ensure that all,
say, balance sheet accounts for a given entity appear once, and only once, on a balance sheet report.

Finally, Starfire contains a number of programs for file maintenance. The transaction file compression program has already been mentioned. There is also a program to roll balances forward to the next financial year.

In short, the Starfire system is a smaller-scale, less flexible, system than its bigger brothers. Provided one wants to do what the system is programmed to do it is also much easier to understand and use.

4.2.3 The Attache microcomputer-based accounting system

Attache4 is one of the half dozen or so most popular PC accounting packages on the market. It is also one of the simplest to use. The package is so small that the executable program files fit on one 360KB floppy disk. Data files consist of one main data file, ATTDATA.DAT, and four index files: customers, products, suppliers, and general ledger accounts. (Attache provides its own index system since DOS provides no file management support.)

*** Figure 4.3 goes approximately here, see next page ***

The Attache system is based around a general ledger and three subsidiary ledgers: accounts receivable, inventory, and accounts payable. The 24 standard transaction types shown in Figure 4.3 invoke pre-coded programs. Twenty-two standard reports may be invoked from a similar screen (options 27 through 52).
<table>
<thead>
<tr>
<th>CUSTOMERS</th>
<th>SUPPLIERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Add, Modify, Delete Customers</td>
<td>15. Add, Modify, Delete Suppliers</td>
</tr>
<tr>
<td>2. Enquire into Customers</td>
<td>16. Enquire into Suppliers</td>
</tr>
<tr>
<td>3. Produce Invoice/Credit Note</td>
<td>17. Bills</td>
</tr>
<tr>
<td>4. Enter Invoice Details</td>
<td>18. Credits</td>
</tr>
<tr>
<td>5. Enter Credit Details</td>
<td>19. Payments to Suppliers</td>
</tr>
<tr>
<td>6. Payments from Customers</td>
<td>20. Supplier Adjustments</td>
</tr>
<tr>
<td>7. Customer Adjustments</td>
<td>GENERAL LEDGER</td>
</tr>
<tr>
<td>PRODUCTS</td>
<td>21. Add, Modify, Delete Accounts</td>
</tr>
<tr>
<td>8. Add, Modify, Delete Products</td>
<td>22. Enquire into Accounts</td>
</tr>
<tr>
<td>9. Enquire into Products</td>
<td>23. Transactions</td>
</tr>
<tr>
<td>10. Receipts from Suppliers (Credit)</td>
<td>24. Enter Authorised Capital Amount</td>
</tr>
<tr>
<td>11. Returns to Suppliers (Credit)</td>
<td>OTHER</td>
</tr>
<tr>
<td>12. Receipts from Suppliers (Cash)</td>
<td>25. End of Accounting Period</td>
</tr>
<tr>
<td>13. Returns to Suppliers (Cash)</td>
<td>26. Change of Date</td>
</tr>
<tr>
<td>14. Product Adjustments</td>
<td></td>
</tr>
<tr>
<td>Page Down for reports and odd jobs</td>
<td></td>
</tr>
</tbody>
</table>

ENTER YOUR SELECTION

Figure 4.3: The first of Attache4's two menu screens
The system is attractive because the program "knows" enough about accounting so that minimal data entry is required. Each of the pre-coded programs is effectively a journal entry template where Attache just fills in the missing data. For example, suppose 100 units of standard inventory item ABC are received from supplier XYZ at a unit price of $10 each. The user selects program 10 from the menu and is presented with a screen that asks for supplier code (enter XYZ or press <F2> to select from suppliers on file), product code (enter ABC or press <F2> for product list), quantity received (100), and unit price ($10). All these entries may be changed until the transaction is accepted (press <F1>). As soon as the transaction is accepted both subsidiary ledgers are updated (inventory and supplier), and the appropriate general ledger accounts are also updated. This journal entry template idea works very simply, and provides a good example of the sort of user interface for the "front end" program discussed in the chapter on Formula Accounting (Chapter 7).

Compared to the two previous systems Attache is very simple. It manages four ledgers, so there are no feeder systems to be considered. Most transactions are pre-programmed, so data entry is user-friendly but flexibility is lost: unusual transactions are difficult to enter. Many of the features of the larger systems are absent. There is little chart of accounts flexibility, no budgeting, no automatic journal entries, no automatic cost allocation, no currency translation, and no historical data stored by month. All reports are pre-programmed. Being a PC system, data are online, which is good. However, selective inquiry/reporting, e.g., of all transactions to a given account in a given period,

---

7 Attache has a table where the user defines which ledger accounts control inventory, accounts payable, accounts receivable, discount, bank, etc.
is limited. In short, this PC-based system is a good example of a simple, user-friendly, unpretentious accounting system suitable for use in a small business. Perhaps the most useful idea to be borrowed from Attache for future TPAS design is its journal entry templates (the programs invoked from the menu in Figure 4.3). These templates make it simple for the user to create complex journal entries by entering minimal data.

4.2.4 **Computer Associate's ACCPAC Plus family of "Accounting" software**

Computer Associates' ACCPAC Plus is a powerful PC-based family of accounting software consisting of the following modules:

* General Ledger and Financial Reporting
* Accounts Receivable
* Accounts Payable
* Inventory Control and Analysis
* Order Entry
* Job Costing
* Retail Invoicing
* Sales Analysis
* US and Canadian Payroll
* Windowing System Manager
* LanPak

Only those modules starred are reviewed here. Under the Windowing System Manager all six applications at the top of the above list may be active on one
machine simultaneously (one presses F10 to step from one task to the next). All ACCPAC modules may be integrated so that, for example, General Ledger accounts entered in any module will be validated automatically during data entry. When all six applications starred above are active simultaneously ACCPAC has about 200 files open, i.e., there are about 30 to 40 files per application. The reason there are so many files is that most data files are indexed (B-tree-based indexed sequential files), and indices are stored as separate files. The LanPak module controls write access to these files so that, under LanPak, a number of users can use the system simultaneously. Compared to Attache, where the program code was less than 360K, ACCPAC is quite large. Program code for the modules starred above occupies about 10MB of disk space.

All ACCPAC modules are accompanied by excellent documentation. The normal structure for each 3-6 cm thick manual is to have a chapter on each of the following: a general outline of the purpose of the module, installation instructions, general instructions for using the package, a tutorial based on sample data included in the package, details of frequently performed tasks, a reference section (organized alphabetically) which describes each feature, and a series of appendices that describe error messages, file structures, and import/export facilities.

---

8 To provide sufficient file buffer space, the user's CONFIG.SYS file is updated automatically during software installation.
The 1990 version of the General Ledger, version 6.0, maintains two years of historical data and one year of budget data for 12 or 13 accounting periods per year. The number of accounting periods per year (12 or 13) is defined at system set-up time. The account identifier may be up to 10 characters long, with two fields: Account, up to six bytes, and Department, up to four bytes. There is no provision for different company codes as there is in M&D’s mainframe GL system. A feature of version 6.0 is that all general ledger data may be exported and imported, very simply, to spreadsheet and database packages. This simplifies preparation of monthly budget figures and their re-entry into the ledger system.

Data are processed in batches, which may be entered directly or retrieved from feeder system modules. Transaction batches may be listed, validated, and modified before posting. (If batches retrieved from feeder systems are modified, the transactions affected are flagged on reports.) Batches may be saved for recurring entry or automatic reversal. They may also be set up to automatically reallocate balances to other accounts. Comments may be attached to batches. Prior periods may be locked to prevent changes from later period batches. For audit trail purposes, transactions can be selected and sorted on date, period, source code\(^9\), posting sequence, batch number, reference, account code, or department code.

---

\(^9\) Source codes serve to define special-purpose journals. Different feeder system modules which create general ledger journal entry batches have different source codes.
Standard reports are available, automatically generated to match the account classes specified by the user during chart of accounts maintenance. However, users may design as many of their own reports as they like, specifying features like rounding, column and row arithmetic, ratios, quarterly and 14-column (period) reporting.

4.2.4(b) ACCPAC Accounts Receivable

The current version of Accounts Receivable, version 5.0, is designed to handle up to 65,000 customers. Open-item and balance-forward customers may be recorded in the system simultaneously. In place of the usual current, 30-days, 60-days, 90-days-and-over classification, aging periods are user-tailorable at report time. For those who need it, the system will calculate interest charges on late payments, and other recurring charges. Customer statements, and mailing labels may be formatted using an in-built text editor. General Ledger journal entry batches are generated automatically by the system. Each invoice may be distributed to an unlimited number of general ledger sales/revenue accounts. Cash flows may be monitored through five general ledger receivables control accounts and five general ledger bank accounts.

4.2.4(c) ACCPAC Accounts Payable

The current version of Accounts Payable, version 5.0, is designed mainly to manage cash payments to suppliers, including automatic cheque printing. By telling the system when you next plan to run your cheque processing it will suggest which vendors should be paid now to maximize discounts taken.
Alternatively, discounts may be ignored, any account payable may be placed on hold, or payments may be forced for the current cheque run. As with Accounts Receivable, aging period reports are user-tailorable (no 30-, 60-, 90-day restrictions), and vendor invoices may be distributed to an unlimited number of general ledger accounts.

4.2.4(d) ACCPAC Inventory Control and Analysis

Version 5.1 of the Inventory Control package (October, 1988) allows users to maintain inventory records for up to 36 locations. Inventory Control maintains records of stock levels, prices, costs, alternate items, and primary and secondary suppliers. Five levels of prices may be defined as either markup or discount percentages on the "normal" price. Markups and discounts may vary from customer to customer, or with the quantity purchased. Sale prices may also be set for specified periods of time. Up to five units of measure may be defined. Where relevant, serial numbers may also be recorded in inventory and shipment records. Costing records that may be maintained simultaneously are moving average, FIFO, LIFO, standard cost, and "user-defined".

The system enables users to enter purchase orders, receipts, shipments, and to process adjustments (from one warehouse to another or shrinkage). Reorder quantities for up to six periods may be defined for seasonal ordering patterns. Sales histories are maintained for the last 13 periods. Standard reports include item valuation, a reorder report, overstocked items, and a physical inventory worksheet. Inventory Control also integrates with the Order Entry module for sales invoicing, and with General Ledger as described below.
ACCPAC Order Entry

ACCPAC's Order Entry module prepares customized orders, picking slips sorted into the order in the warehouse, packing slips (with estimated shipping weight calculated) and invoices with special pricing for certain customers or categories of items. Special instructions and comments may be included on the shipping documents, some to appear on picking slips, other to appear on invoices. Items are automatically backordered if out of stock. Invoices generated by the Order Entry module automatically update the inventory records of the Inventory Control module. They also generate batches of transactions to update both the General Ledger (Sales Revenue, Accounts Receivable Control, Inventory, Cost of Sales, Sales Commissions, Discounts, Taxes Payable, etc.) and individual customer records in the Accounts Receivable module. (Inventory Control files store pricing information, but Inventory Control does not prepare sales invoices.) Because it was designed for the North American market, tax tables may be set up to handle different tax calculations for different geographic areas for invoices. Sales commissions are calculated based on price or margin. Sales history reports include analyses by product, area, and salesperson.

ACCPAC Job Costing

The current version of Job Costing, version 5.0, is designed to track costs incurred on different jobs or projects. At the planning stage jobs are divided into "phases", and phases are divided into "categories". Budget costs are prepared for each job, phase, and/or category. (If revised, initial estimates and subsequent revised costs are all stored.) Costs are charged to categories as they
are incurred. The total amount billed, payments received, costs incurred, and
cost to complete the job are available at any time. Percentage completion
figures are also available automatically, with costs over or under plan high-
lighted. Past job information may be saved to facilitate setting up future jobs.
Revenue and expenses may be recognized at any time in the accounting cycle,
e.g., on some jobs revenue may be recognized on completion of a contract, on
others, percentage completion may be used. Job Costing integrates with
ACCPAC's General Ledger, Accounts Payable, Accounts Receivable, Payroll
(US and Canada only), and Inventory Control modules.

4.2.4(g) Summary: The ACCPAC Plus Family

ACCPAC is a good example of a reasonably powerful suite of accounting
packages. Scanning the huge list of ease-of-use features mentioned in the last
few pages it is apparent, however, that the accounting model underlying
ACCPAC is extremely simple. Orders, invoices, inventory, accounts receivable,
accounts payable, and cash payments are the simplest of transactions to account
for. Evidently, the objective of designers of the ACCPAC packages has been to
simplify record-keeping, and to assist with management of current assets and
liabilities. What is missing is any attempt to use computer technology to
improve the quality of income measurement, which is the prime objective of
this thesis.
The final TPAS package to be reviewed in this section, Section 4.2, is MSA's Fixed Asset system. Fixed Asset systems are of interest because they are the only modules in most TPAS (with the exception of the currency translation features of the general ledger systems) that confront the problems of asset valuation, and therefore income measurement. The material in this section is based on a manual published in 1985 [MSA, 1985] when FASB 33 was still in force in the US. The striking feature of this system, as with all the companion modules available from MSA and its mainframe competitors, is complexity.

A Fixed-asset TPAS is basically used to maintain a record of non-current assets, calculate depreciation expense for accounting reports, and perhaps most important, to take advantage of tax legislation to maximize tax depreciation. The MSA Fixed Asset system integrates with MSA's Accounts Payable, General Ledger, and Project Tracking systems. The link with Project Tracking enables a user to combine information about Fixed Assets and Development Projects on one report. The MSA system provides seven sets of "books", the first for the normal accounts, the second for the tax books, leaving five others at the discretion of the accountant. An Australian subsidiary of a US firm might use Book 3 for US reporting.

Responsibility for each asset is defined by a 14-level control code, e.g., Company, Subsidiary, Division, Plant, Department, Cost Centre, Process, Building, Floor, Account, Category, Equipment Class, General Ledger Distribution, and Project. The top two codes and an asset number uniquely
define each asset, so the remaining codes are just independent selection/aggregation criteria for reporting purposes. Standard information about most assets includes asset description, producer/manufacturer, model name, asset serial number, acquisition date, cost, year-month depreciation commences, estimated life in years, depreciation method, date retired, salvage value, etc.

The system designers have addressed most of the practical problems of dealing with fixed assets. Large assets, like buildings, may be viewed as aggregates of many components. Consistent with the cost-based emphasis of GAAP accounting, selling or adding to a component changes the overall value of the aggregate asset. Major repairs are treated the same way. Fractions of "an asset" may be sold. For instance, a hotel with an asset "340 writing desks" may sell off, say, 20% of its desks, and the system will reduce cost, quantity, and accumulated depreciation by 20%. For depreciable assets, standard depreciation rules, e.g., sum of years digits, for the US, UK, Canada, and Australia, are built-in, as are detailed rules for calculating US taxable depreciation. If all else fails the user can define a depreciation table and specify percentages of cost for each year. For US companies the system automatically switches tax depreciation methods at the right time to maximize the deduction. Part-year depreciation methods for the first year of ownership include half year, modified half year, periods after, full year, actual months, and no depreciation. For non-depreciable assets there is a US property tax compliance report.
For leased assets, the system "automatically calculates and produces the reports needed to meet internal lease accounting regulations, including those in the US, UK, Canada, and Australia." [p.28] This includes distinguishing between capital and operating leases, discounting a schedule of future lease payments at a user-defined interest rate, reporting lease obligations and amortizing the corresponding asset, and calculating gains or losses on retired or cancelled leases.

For inflation accounting, the system again claims to meet the requirements of the US, UK, Canada, Australia, New Zealand, and The Netherlands. This includes all the different nuances of SFAS 33 [FASB, 1979], SSAP 16 [ASC, 1980], Canada's CCIA Section 4510, and so on. Quoting from the manual, the system provides three methods of asset valuation:

- **Direct valuation** -- You supply the gross current cost of the asset each year or whenever the value changes.

- **Indexing** -- Indexing is the application of indices to the base cost to determine current cost.

- **Direct valuation with indexing** -- This method is a combination of the two preceding it. At some point in the life of the asset, the gross basis may be restated to a value representing an appraisal, catalogue price, and so on. The restated value is indexed from that point." [MSA, 1985, p.23]

There seems to be no practical limit to the number of price-index series the user may define and maintain. Tables may have either monthly or annual values, the latter in average or end-of-year currency units. For compliance with SFAS 33, gains and losses above and below the general level of price changes are segregated and reported.
Summarizing, designers of this the Fixed Asset system have had to confront many of the thornier problems of accounting: special books for financial reporting and for tax, assets that are aggregates or parts of other assets, the plethora of rules for depreciation, lease accounting, and inflation accounting. These issues are, of course, different in the different countries where MSA sells it software. The system generates journal entries that feed automatically into the general ledger. At present this would not include journal entries for inflation accounting (CCA is not mandatory), but obviously it could.

4.3 Chapter Summary

The results from the questionnaires reported in Section 4.1 indicate that computer-based general ledger systems are widely used in our community. Most companies installing systems today tend to buy packaged software, rather than have it custom-built. The general ledger package is the most popular, but many firms also purchase other modules, like accounts payable, and accounts receivable, invoicing, and inventory, to capture details of transactions and feed them to the general ledger. Figure 4.1 therefore describes the structure of many of today’s computer-based accounting systems.

At present there are good commercial reasons why such feeder systems and general ledger packages are developed and sold as separate modules. However it does not require too much effort to imagine a future system where the data from all the feeder systems were stored in, say, a group of distributed databases, with conventional accounting information being extracted more-or-less automatically.
The detailed review of accounting software in Section 4.2 showed that today's TPAS are designed to assist management with detailed record-keeping and reporting, not to assist external users of financial reports. Today's large general ledger systems are huge repositories of data, in some cases storing something like three years of monthly actual, budget information, and possibly, revised-budget, information, for each of sometimes hundreds of thousands of accounts. In addition to the general ledger, smaller integrated systems like Attache and ACCPAC manage subsidiary ledgers like Accounts Receivable, Accounts Payable, and Inventory, (as well as Purchase and Sales Orders in the case of ACCPAC) that are clearly designed to simplify record-keeping for current assets and liabilities, not to improve income measurement. Apart from the fixed-asset system reviewed in Section 4.2.5, none of the systems studied showed any interest in accounting for changing prices (though, of course, expectations of inflation can be built into budgets). Asset values could be restated using techniques like those in the fixed-asset module, but they are not widely used, and there seems to be little demand for such features.

Perhaps the single, most obvious conclusion to be drawn from this chapter's review of computer-based TPAS in practice is that developments in computer-based accounting systems have been aimed at making day-to-day recording and reporting easier to do, not at improving or changing the underlying system of ledger accounts with (normally historical cost) money balances.

(9,200 words)
Chapter 5: Computer-based Accounting: Theory

Academic papers on the design of computer-based accounting systems can be counted on one hand. The main papers are Everest and Weber [1977] and McCarthy [1979, 1982], which suggest some normative guidelines about how databases for accounting systems should be designed, and Weber [1986] which tests McCarthy’s REA model [1982] empirically. Of more concern, the academic literature has largely been ignored by practical software developers [Weber, 1986, p.499] and text-book writers. This chapter reviews the sprinkling of journal articles that do exist, Ijiri’s proposals for Multi-dimensional Accounting [1966, 1975] (which have influenced a number of papers on computer-based accounting), and the text-books on Accounting Information Systems (AIS). It begins with the AIS text-books.

5.1 Textbooks on computer-based accounting

All of the dozen-or-so texts on Accounting Information Systems include chapters describing a typical suite of computer accounting systems, e.g., sales and accounts receivable, purchases and accounts payable, payroll, fixed assets, production planning, and of course, a general ledger system. Table 5.1 shows that most texts devote about 20% to 30% of their pages to such descriptions. None of those descriptions is as complete as Chapter 4 (if they were, Chapter 4 would not have been necessary). Of course, most texts devote more time to organizational issues like internal control and document flow that are important, but not relevant for this thesis.
A search of chapter bibliographies of the seven texts in Table 5.1 shows hardly any reference to the academic literature; most of the few references are to software supplier manuals. By devoting hundreds of pages to descriptions of TPAS, textbook writers are recognizing that their importance. In addition, by referring only to software vendor manuals they are signalling that academic research in the area is not helpful. As with Chapter 4 in this thesis, textbook descriptions tend to say "this is how the world is", or "this is how accounting is done". They make little attempt to ask whether what is done is in any way optimal, or whether computer-based accounting could be improved by some alternative information system design.

Table 5.1: Textbook descriptions of computer-based accounting systems

<table>
<thead>
<tr>
<th>Accounting Information Systems Textbook</th>
<th>Total pages</th>
<th>Descriptions of computer-based system cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>%</td>
</tr>
<tr>
<td>Bodnar and Hopwood [1990]</td>
<td>1054</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21%</td>
</tr>
<tr>
<td>Cushing and Romney [1990]</td>
<td>870</td>
<td>215</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25%</td>
</tr>
<tr>
<td>Davis, Alderman, and Robinson [1990]</td>
<td>760</td>
<td>310</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41%</td>
</tr>
<tr>
<td>Hicks and Leininger [1986]</td>
<td>748</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17%</td>
</tr>
<tr>
<td>Moscove and Simkin [1990]</td>
<td>774</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2%</td>
</tr>
<tr>
<td>Page and Hooper [1987]</td>
<td>738</td>
<td>190</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26%</td>
</tr>
<tr>
<td>Wilkinson [1989]</td>
<td>702</td>
<td>196</td>
</tr>
<tr>
<td></td>
<td></td>
<td>28%</td>
</tr>
</tbody>
</table>

One other (non-AIS) textbook not listed in Table 5.1 is Kraushaar and Sullivan [1986], which contains what is probably the best description of present-day computer-based accounting systems in any text. Half the 30-page chapter
describes general ledger systems. The other half covers Accounts Payable, Accounts Receivable, Payroll, and Fixed Assets. Kraushaar and Sullivan begin their chapter by saying that financial information systems "have evolved at an extraordinary pace during the 1970's through the mid-1980's" [p.6-3]. This is certainly consistent with the findings reported in Chapter 4. The systems surveyed in Section 4.1 were all installed in the last 15 years, and it seems doubtful that anyone could predict what their replacements will look like in 15 years time.

5.2 Journal articles on computer-based accounting

The academic literature on computer-based accounting begins with a few tentative efforts, like Eaves [1966] and Mathews [1967], that explored efficient ways to use the then new, and very expensive, computer technology for maintaining a general ledger. In the early to mid-1970's, there was a flutter of interest when Colantoni, Manes and Whinston [1971], Lieberman and Whinston [1975], and Haseman and Whinston [1976], all at Purdue, speculated on ways of applying the then new database technology to accounting systems. Lieberman and Whinston [1975] suggested that database technology might be used to implement an Events Accounting system [Sorter, 1969]. Haseman and Whinston [1976], suggested that Ijiri's Multidimensional Accounting [1967, p.110] could be used as the basis of a computer-based accounting system. The Purdue papers are full of diagrams of simple data structures for representing trees. They are more about Lisp programming and network database design than about computer-based accounting. Reflecting the computer science emphasis of these papers, two of the authors of those papers went on to form a successful
commercial software company that developed the database management system, Knowledgeman, and more recently the expert system shell, Guru. However, their papers are of little interest today.

In the late 1970's a few accountants continued to take novel ideas from the rapidly growing database literature and to apply them to accounting. Everest and Weber [1977] took Codd's [1971, 1972] ideas on relational databases and normal forms and attempted to design relational databases for accounting. One comment they made heralds an issue important to this thesis. Everest and Weber [1977] say:

database management systems deal with real entities, such as jobs, projects, departments, people, and parts. However, the accountant deals with many "artifacts". Charts of accounts are useful taxonomies, classification schemes or naming conventions rather than real entities. Database management systems typically handle these artifacts in an awkward and unwieldy manner. [p.342]

Of course, the last sentence is not true. Charts of accounts and journal entries can be modelled very readily in a database system. The problem that Everest and Weber were trying to identify is one of different levels of abstraction. Accounting systems are much more abstract interpretations of real-world events that may also be described in terms of departments, people, and parts. As will be discussed shortly, McCarthy [1982] encountered this abstraction issue too. He appears to have been the first to suggest that double-entry accounting
should be regarded as a view; a view derived from an underlying conceptual schema based on resources, agents\(^1\), and events.

McCarthy's PhD thesis [1978] applied Chen's [1976] Entity-Relationship (E-R) data modelling methodology to design a conceptual schema for Events Accounting in a small trading firm. He summarized the key points from his thesis in an article in *The Accounting Review* entitled "An Entity-Relationship View of Accounting Models [McCarthy, 1979]. That paper was the original stimulus for the present thesis, and most of the remainder of this section is devoted to McCarthy's work.

*** Figure 5.1 goes approximately here, see next page ***

McCarthy [1979, Figure 6, p.675] presents an E-R model, reproduced as Figure 5.1, for the trading firm modelled in his thesis. Commenting on this model, which seems very simple when compared to the databases of real-world software, McCarthy says:

> The data model developed to this point for a small retail enterprise certainly differs radically from the traditional accounting paradigm generated by using a chart of accounts and double-entry bookkeeping. [p.681]

His point is that his model in Figure 5.1 contains information about (a) events like purchases and sales, cash payments and cash receipts, changes in capital, (b)

---

\(^1\) These agents are people and firms who may cause events to happen. They are agents in the sense of Agency theory [Jensen and Meckling, 1976].
FIGURE 6
E-R Diagram for the Entire Retail Enterprise

CUSTOMER

PURCHASE

VENDOR

INVENTORY

STOCKHOLDER

GENERAL & ADMINISTRATIVE SERVICE

EQUIPMENT

EQUIPMENT ACQUISITION

SALE

ORDER

CAPITAL TRANSACTION

CASH RECEIPT

CASH DISBURSEMENT

CASH

PERSONNEL SERVICE

EMPLOYEE

payment for

supplier of

allocate cost of

flow of

payment for

flow of

received from

made to

line item

line item

line item

payment for

line item

payment for

partner to

payment for

flow of

payment for

flow of

line item

line item

line item

payment for

CASH PERSONNEL SERVICE
people or other firms who buy from, sell to, or otherwise interact with the firm, and (c) resources managed by the firm, e.g., stocks of cash, inventory, equipment, and although McCarthy does not mention them, liabilities. Nowhere in these lists is there any mention of ledger accounts or debits and credits so McCarthy concludes that Everest and Weber [1977] were correct in describing charts of accounts, and double-entry bookkeeping, as accounting artifacts not easily modelled in database systems.

McCarthy goes on to devote some pages to claiming that the E-R approach to data modelling is more useful than Codd's relational decomposition approach used by Everest and Weber [1977]. Today the E-R approach is viewed as one of many competing methodologies for data modelling/conceptual schema design. Each has its own strengths and weaknesses, none being the whole answer [Verrijn-Stuart, 1986, pp.12-17]. Over the next four years to 1982 McCarthy tested and refined his ideas. McCarthy [1980] emphasizes that there are many information users, including non-accountants, within an organization, and presents some simple E-R models to suggest how information in a corporate database could be organized for sharing. McCarthy and Gal [1981] describe a CODASYL implementation of the Figure 5.1 database.

5.3 McCarthy's REA Accounting Model

McCarthy's 1982 paper presents (i) a clearer idea of the relationship between conventional accounting and the organization's underlying database than prior work, and (ii) a meta-model of the facts that must be recorded in a database if accounting reports are to be generated.
5.3.1 Relationship between conventional accounting and the organization's underlying database

McCarthy's Figure 2 [1982, p.560] suggests that conventional accounting should be regarded as an external schema, or view, derived from the underlying enterprise conceptual schema. He states:

It is a primary contention of this paper that the semantic modeling of accounting object systems should not include elements of double-entry bookkeeping such as debits, credits and accounts. [pp.559-560]

Reinforcing this idea he says:

For example, "cost of goods sold" and "sales revenue" are two distinct accounts. In a modeling sense, however, the two accounts represent attributes of the same entity set "sale events". [p.561]

It is probably fair to say that the real-world accounting systems reviewed in Sections 4.1, and 4.2 are evolving into a complex implementation of this perspective. At present each feeder system in Figure 4.1 has its own file system, and special interface programs often have to be run to generate and format journal entries for the general ledger, but even at this early stage of development of "database accounting" the general ledger can already be regarded as just a view extracted from the underlying feeder systems. A trend at the moment is for large-system software vendors to offer DB2 versions of their software. Whether it will ever be practical, or even desirable, for all

\(^2\) IBM's mainframe relational database package

5 - 7
feeder systems to store their data in one large DB2 database, is unclear. But if it were, the tantalizing next step would be to try to define the entire general ledger as just a view of the feeder system data.

5.3.2 The REA Accounting model

The second, and, from the title and tone of his paper, more significant contribution of McCarthy [1982] is the meta-model which McCarthy called the Resource-Events-Agents (REA) Accounting Model. Features of this model, reproduced here as Figure 5.2, will now be discussed.

*** Figure 5.2 goes approximately here, see next page ***

Referring to Figure 5.2, there is no question that Economic Events may be linked via stock-flow relationships with Economic Resources. The idea that economic events involve exchanges of economic resources is a fundamental building block in many accounting systems, e.g., Ijiri's Multidimensional Accounting [Ijiri, 1975]. This same idea is also the basis of the Resources and Exchange Events accounting model developed and presented in Chapter 9 of this thesis.

As discussed below, Weber [1986] believed he found a problem with the control relationship between Economic Events, Economic Agents, and Economic Units, at the right-hand side of Figure 5.2, but Weber also suggests a solution.
Figure 5.2  Entity-Relationship diagram of McCarthy's REA Accounting Model, from McCarthy [1982], p.564, Figure 5.

**Figure 5**

The REA Accounting Model

(a) Entities and relationships

(b) Role declarations

- **Participating Entity Sets**
  - stock-flow
  - duality

- **Roles**
  - stock
  - flow
  - increment
  - decrement
  - superior
  - subordinate
  - exchange transaction
  - outside party
  - inside party

- **ECONOMIC RESOURCE**
- **ECONOMIC EVENT**
- **CONTROL UNIT**
This leaves the most distinctive part of McCarthy's REA model, the duality relationship between Economic Events, to be discussed. McCarthy [1982, p.562] defines duality as follows:

Duality relationships link each increment in the resource set of the enterprise with a corresponding decrement [Ijiri, 1975, Ch.5]. Increments and decrements must be members of two different event entity sets: one characterized by transferring in (purchase and cash receipts) and the other characterized by transferring out (sales and cash disbursements). The abstract notion of duality is described in detail by Mattessich [1964, pp.26-30]. [p.562]

Since this duality concept is a key difference between McCarthy's REA model, and the Resources and Exchange Events model presented in Chapter 9, it is essential to clarify precisely what is meant by "duality". It turns out that, notwithstanding the last quotation above, McCarthy's concept of duality is not the same as that used by either Ijiri or Mattessich. The duality concept used by both Ijiri and Mattessich has three attributes: (i) one event, (ii) dual resources being exchanged, and (iii) the same value for the resource given up, and the resource acquired. Mattessich defines his concept of duality as follows:

we may interpret the duality principle as the assertion that a transaction or flow has two dimensions: an aspect and a counter-aspect (to avoid the terms input and output which have too concrete a flavour, or the terms debit or credit which have too strong a flavour of the technical recording process). More precisely, the principle asserts that there exist economic events which are isomorphic to a two-dimensional classification of a value within one set of classes. [p.27] (emphasis in the original)

3 The shadowing on the last five words flags some key words in the argument. They were not in the original. Shadowing in the next two quotations has the same purpose; it was not in the original text.
It is hard for mere mortals to make much sense of such a definition. But one can see that Mattessich refers to "a transaction" (one event), some sort of "aspect and counter-aspect", of which the accountant's use of debit and credit is an example (dual resources), and "a value" (one value).  

Ijiri, in his Chapter 5 "The Axiomatic Structure of Historical Cost Valuation", distinguishes between classificational and causal double-entry: 

there are two entirely different reasons why debit should equal credit. One is that they are both based on different classifications or descriptions of the same object. We call this type of double-entry classificational double-entry. ... The other case of double-entry is what we may call causal double-entry, where the value of an increment (debit) is set equal to the value of a decrement (credit), as in (Dr.) Inventories $100: (Cr.) Cash $100. ... This entry clearly involves two different resources, cash and inventories. ... Debit equals credit apparently because of the historical cost principle, namely the value of the increment is set equal to the value of the decrement in an exchange.

[p.81]

As with Mattessich, it is apparent that Ijiri's duality involves one event ("an exchange"), dual resources ("two different resources"), and one value ("the value"). However, McCarthy's notion of duality is different: it involves two events. His exact words in the quotation given earlier (citing Ijiri and Mattessich) are "two different event entity sets". Thus McCarthy's concept of

---

4 The use of one value for both the input and output resources is most apparent in Mattessich's later example [p.113] of an economy-wide input-output matrix as an example of duality. In such a matrix, a number in one cell simultaneously specifies the value of what has been used up, and the value of what has been produced.

5 Ijiri [1975] actually uses the term "duality" in a different sense in Chapter 5 to his use in Chapter 4. In Chapter 4 he talks of the dual values of an item of inventory, its buying price, and its selling price. In chapter 5 he uses "duality" in the sense described in this thesis.
duality is his own, not Ijiri's, nor Mattessich's. His usage is, however, consistent with much of Ijiri's discussion in Chapter 5 of causal chains of events. For example, in his Chapter 4, Ijiri [1975] defines a causal chain as follows:

Suppose that during period 1 a retailer purchases merchandise G for $2 cash, denoted by M, and during period 2 he sells it for $3 in cash, denoted by M'. The following chain of exchanges, or causal chain results:

\[(3.1) \ (\$2) \ M \rightarrow G \rightarrow M' \ (\$3) \] [p.62]

Clearly this chain describes two events, in two time periods. McCarthy seems to have equated "causal chain" with "causal double-entry".

Having established that McCarthy's duality (i.e., "causal chaining") is a link between two events, not an exchange of resources within one event, one must now ask whether it deserves its prominence in McCarthy's REA meta-schema, Figure 5.2. On the positive side, most sales invoices quote the customer's purchase order number, and credit notes often quote the prior invoice that the credit offsets. More examples may be found in Figure 5.1 above, where McCarthy has ensured that all seven event entity types (left to right: Order, Sale, Cash Receipt, Purchase, Capital Transaction, Cash Disbursement, and Equipment Acquisition) are linked to one, and only one, other event entity type. These are simple, useful, instances of related sequences of events.

However, on the negative side, causal chains can become hopelessly entangled. Convincing evidence of practical problems with causal chains is found in conventional accounting treatments of both Accounts Receivable and Inventory:
(a) Accounts Receivable

For years, accountants have debated about the relative merits of open-item versus balance-forward Accounts Receivable systems. Under an open item system invoices are listed separately, and when cash is received it is supposed to be applied to specific invoices. Likewise for credit notes. Under a balance-forward system the Accounts Receivable balance is just a record of the amount of money owed at any time. Invoices add to it, cash receipts reduce it, credit notes reduce it. In a balance-forward system there is no attempt to link cash and credit notes to invoices as McCarthy would have us do. Both systems work; they both record the customer's account balance, and they both maintain transaction histories to explain the balance. When dealing with large numbers of credit notes, or small traders who want to make part payments, e.g., "$1,000 today, and the balance next month", the balance outstanding becomes all-important, and the balance-forward method is simpler to use.

(b) Inventory flow assumptions: LIFO and FIFO

Techniques like LIFO and FIFO are used by accountants world-wide to value inventory. If causal chains were maintained as McCarthy suggests, LIFO and FIFO would not be necessary because each sale event would be linked back to some prior purchase-of-inventory event. The fact that LIFO and FIFO inventory flow assumptions are in such widespread use is particularly strong evidence of the practical problems that maintenance of causal chains entails.
In short, it seems that whenever accountants find the causal chaining of events too complex, they define a balance-forward entity such as Accounts Receivable, or Inventory, to add to, and subtract from. For instance, one could easily define a balance-forward entity called Accounts Receivable, between McCarthy's Sale and Cash Receipt event types near the bottom left-hand corner of Figure 5.1.

How do balance-forward entities fit into McCarthy's REA meta-model in Figure 5.2? The answer is that balance-forward entities are stocks of resources. Each of the five resource entity types in Figure 5.1 (left to right: Inventory, Cash, General and Administrative Service, Equipment, and Personnel Service) has a quantity attribute that rises and falls over time. It is always true (tautologically) that changes in quantities can be explained by some kind of flow (a step function of time like an event, or a smoother function of time like evaporation). Thus, provided one treats McCarthy's REA Economic Event, Figure 5.2, as a general kind of flow (which is consistent with his definition on page 562), the stock-flow link to at least one Economic Resource is always true. However, it is only sometimes true that one will want to link one kind of flow to some earlier or later kind of flow. As illustrated above, McCarthy's duality link between Sale and Cash Receipt, in Figure 5.1, can be removed and replaced by a new Resource type, Accounts Receivable. Thus, McCarthy's inclusion of "duality" relationships in his REA meta-schema as something that must always be modelled, is too strong. Sometimes these links between events are useful, but not always.

In the only published attempt at empirical evaluation of REA, Weber [1986] made an in-depth study of what he calls "Wholesale Distribution Software", i.e.,
order-entry software used by wholesalers. Weber sought, amongst other things, to test the match between the data structures in the 12 packages he examined and the REA model. Before going on, there are two warnings about generalizing from Weber's paper. First, orders from customers for inventory are only a very small part of an accountant's universe of discourse. There is no logical basis for wider generalizations. Second, Weber's clustering approach to mapping what he found onto REA is, in this writer's opinion, unsound. Under a strict interpretation of REA, the only entity types allowed in an REA schema are Resources (R), Events (E), and Agents (A), and their subtypes. Now, Weber's Figure 4, pp.510-511, reproduced as Figure 5.3, includes all manner of entity types, like inventory-price categories, warehouses, messages, and non-stock line items, that are not subtypes of R, E, and A, in the clusters labelled "Economic Resource", "Economic Event", and "Economic Agent". In this writer's opinion, Weber's clustering is inconsistent with the notion of a subtype as used in data modelling, e.g., see Smith and Smith [1977], and therefore misrepresents the specification of REA.

*** Figure 5.3 goes approximately here, see over page ***

These warnings aside, there are two lessons to be learned from Weber's study of the adequacy of the REA model. First, Weber's problems with the clustering invite an alternative solution. It is suggested that a reasonable alternative test of the appropriateness of REA to any database design, not just wholesale

---

6 Subtypes are acceptable because the R, E, and A types are defined as root nodes in their respective generalization hierarchies, as shown in McCarthy's Figure 1 [1982, p.559].
distribution software, is to see if an REA **view** could be derived from the underlying database. To apply this alternative test to Weber's schema, one treats all of his entity types (rectangles) and relationship types (diamonds) in Figure 5.3 as relational tables, and asks "Is it possible to derive eight tables that correspond to REA, Figure 5.2, as **views** of Figure 5.3?" If so, then the Figure 5.3 schema could be said to conform to the requirements of REA. If not, then the schema does not conform to REA, and REA must be rejected.

Applying the test to Figure 5.3, there seem to be only two event entity types, a Sales Order and a Contract. Consistent with REA, both are linked to Resources, on the one hand, and to Agents and Salespersons (via control relationships) on the other. However, the apparent absence of McCarthy's duality relationships on either event type is a problem. There may have been links in the software, not shown in Figure 5.3, but Weber specifically mentions there was no "duality" relationship for the Contract event [Figure 5, p.512]. If empirical evidence of duality relationships cannot be found, the duality relationship part of REA should be rejected. The Resource and Exchange Events (REE) model introduced in Chapter 9 models the same economic relationships without recourse to duality.

Weber's second finding [pp.512, 516-7] was that REA's ternary "control" relationship is better represented by two binary relationships, one indicating entities outside the firm involved in the event, and one identifying people inside the firm. This criticism will not be examined further, since agency control relationships need not be modelled in REE.
5.4 **Chapter Summary**

This review of the academic literature on suggestions for improvements to computer-based accounting has identified two main contributors. The first is Ijiri, whose work on a theory of accounting measurement was not aimed at computer-based accounting at all, but which is relevant, nonetheless. Ijiri provides some fundamental insights into the process of, and reasons for, account-keeping. In particular, his notions of causal double-entry and Multidimensional Accounting provide foundations for the REE model described in Chapter 9.

The second main contributor is McCarthy, whose work was discussed at length in this chapter. Probably, McCarthy's most important suggestion is that the conventional debits and credits accounting model should be regarded as an external schema, or view, derived from a database describing resources, events and agents. The present trend observed in Chapter 4, for more integration between feeder systems and the general ledger, is rapidly making that suggestion a reality.

The value of McCarthy's REA accounting model [1982], reproduced as Figure 5.2, is less clear. It would seem useful for an accounting system to maintain a record of all the resources under a firm's control, and the events that led to the present state of affairs, but REA's duality relationship between event types is a problem. It can result in hopelessly tangled causal chains of events that are frequently avoided in practice by using intermediate "balance-forward" object types. Any system of accounting that uses balance-forward object types like
inventory, accounts receivable, and accounts payable without explicit links between events that caused the balance to rise and events that caused the balance to fall is evidence that McCarthy's duality is not required for accounting systems to work.

Perhaps the greatest problem with McCarthy's REA meta-model is that it was defined as a normative statement about how accounting systems should be designed, but without any suggested tests or purpose. Had it been defined as a hypothesis to the effect that REA relationships would always be found in "good" accounting systems it would have been testable. Alternatively, if it had been developed for some specific range of purposes it would be possible to test to see if it would work as predicted.

(4,100 words)
One text that presents a novel computer-based accounting system design is Merrett [1984]. In what is basically a computer science book about relational databases and file design, Merrett constructs what he calls a general accounting/financial system in Chapter 6, pp.335-364. The design might best be described as a practical approach to matrix accounting, and consists of four base relations:

TYPE (ACCT, ACTYPE, ADESCR)
TRANS (ACCTCR, ACCTDB, TDESCR)
BALANCE (DATE, ACCT, BAL)
SPREAD (PEREND, ACCTCR, ACCTDB, AMOUNT)

from which revenue, expense, and income statements may be generated. TYPE stores the account code, its type, and description. TRANS limits the allowed types of journal entries, by defining allowed debit-credit pairs, and the transaction description. BALANCE contains opening balances of accounts at specified dates. SPREAD records the net transaction amount for each debit-credit cell in the matrix during some period ending at PEREND. Merrett suggests that SPREAD could be derived from a more detailed history of all transactions if so desired. He also says that SPREAD may be thought of as the result of applying an operator, SURFACE, to a hidden table

SPREADC (PEREND, ACCTCR, ACCTDB, AMOUNT, CHRONOL)

where CHRONOL is a date-time stamp that allows historical tuples, e.g., changes due to error correction, to remain in the relation SPREADC. (The operator SURFACE merely selects the latest (chronologically) tuples for any given PEREND, ACCTCR, ACCTDB.) From the point of view of this thesis the weakness of Merrett’s system is that it assumes the accountant will define the accounts to be debited and credited, and the money value involved. In this sense the system is entirely conventional. There is little scope for alternative valuation systems like current-value accounting.
Part B: An Architecture for Computer-based Accounting Systems
Building on the ideas presented in the last four chapters, the purpose of this chapter is to present an architecture for future computer-based accounting systems. In Sections 6.1 and 6.2 the arguments of the last four chapters are assessed and two fundamental propositions, called Design Principles, are formulated as the basis for further design. Section 6.4 accepts these Design Principles as given and, using Figure 6.3 as a guide, asks how data in the corporate database might best be structured to achieve the goals of Design Principle 1. Two independent, but not mutually exclusive, solutions are suggested. These are outlined in the latter part of Section 6.4, and developed in the next four chapters of the thesis.

6.1 Design Principle 1: Inflation/Current-value Accounting

It is worth trying to build a cost-competitive, general-purpose, computer-based accounting system that is able to generate reports in inflation-adjusted/current-value terms.

Design Principle 1, stated above, was formulated after careful consideration of the arguments presented in Chapters 2 and 3 and additional evidence, presented below, concerning interest in inflation accounting in times of high inflation. Note that Design Principle 1 is a general endorsement of inflation accounting. No decision in favour of any specific form of inflation accounting, e.g., CPP, CCA, CoCoA, or CC/CPP accounting, is implied or intended.
Chapter 2 provided a review of normative accounting theory. First, three Objectives of Accounting were identified. Second, income measurement was singled out as the most important task undertaken by accountants. Third, the nature of the source data needed for income measurement was considered. Fourth, because of large measurement error in systems based on independent valuation of net assets at the start and end of a period it was suggested that income is likely to better measured, i.e., measured with less error, by matching costs against revenues. Fifth, five major systems of accounting (for asset valuation and income measurement) were summarized in Section 2.4. No attempt was made, however, to argue that any one of these systems was better than another. Consideration of the merit of the five systems was deferred until empirical evidence from the recent price-change accounting experiments had been examined in Chapter 3.

Chapter 3 reviewed the empirical literature on inflation accounting, particularly capital markets research into the impact of SFAS 33 (FASB, 1979) and SSAP 16 (ASC, 1980). The evidence shows that market analysts in the US and the UK were unimpressed by standard setters' attempts to create a system of inflation accounting: in essence, CCA reports added nothing to historic-cost-based GAAP reports. For many practicing and academic accountants this evidence was a death sentence for inflation accounting.

However, it will now be argued that the empirical evidence from the early 1980s suffers from a data clustering problem: it only applies to relatively low levels of inflation, say, in the range from zero to twenty percent. At higher levels of
inflation the normative arguments are supported by the empirical evidence presented below.

6.1.1 Justification of Design Principle

This section is devoted to tests of the hypothesis that interest in inflation accounting has tended to wax and wane with the level of inflation a country is experiencing\(^1\). Four cases of high to very high inflation in different countries this century are examined\(^2\). In each case there were calls -- not just one or two voices but a widespread clamour and in two cases government legislation -- for improvements to historical-cost accounting. It is concluded that the problem of accounting in times of changing prices is a real one, and that the lack of interest in SFAS 33 [FASB, 1979] and SSAP 16 [ASC, 1980] is due to the relatively low levels of inflation that existed during the test period.

Once it is accepted that inflation accounting is essential in times of hyper-inflation, a more interesting question follows: What level of general inflation that must exist before inflation accounting reports become more useful than

---

\(^1\) Tweedie and Whittington (1984) offer a similar explanation of interest in inflation accounting. They say:

During the post-war period, interest in inflation accounting in both the USA and the UK proceeded continuously but at a level which tended to vary with the immediacy of the problem, as indicated by the rate of inflation. Mumford (1979) provides an interesting survey of the post-war inflation accounting debate in the UK interpreted in this manner. [1984, p.35]

\(^2\) Apart from Israel, for which data were hard to obtain, these are the main cases of high inflation in countries with reasonably sophisticated accounting this century. Periods of inflation prior to this century, e.g., during the Napoleonic wars, were ruled out because of the relative unsophistication of accounting practices of the time.
Historical Cost? Data are limited in this area, but experience in Brazil, discussed below in Case 3, offers some answers to this question.

Case 1: The German Hyperinflation, 1919-1924.

Geldmacher [1920, p.364], translated by Clarke and Dean [1989, p.295], provides a vivid picture of the accounting problems in Germany during the early stages of the inflationary period, in 1919:

Annual profits are growing, as evidenced by many balance sheets! Mere tricks with figures!... Commercial profit accounting has gone wrong in this day and age... Especially since November 1919 many business people have been complaining almost ceaselessly that the money they are getting their hands on through the sale of their goods and products is often far less than what they have to invest, in immediately replacing the goods sold...[then, in the context of the 3,000 marks reported profits of a piano dealer, he decries] Had not the dealer sold his ten pianos, he would not have made any profit, but he would now own ten pianos of the same quality [and valued at 100,000 marks], instead of only one [valued at 10,000 marks].

Conditions were much worse by 1923. Sweeney [1964, p.xxi] says

the German inflation was the most extreme in history, eventually reaching a trillion to one by the close of 1923. People there went shopping with market baskets full of paper marks, banks set up booths on street corners so that merchants could convert their cash receipts immediately into dollars,...

In fact, inflation was such a major problem, not just for accountants, that in desperation the German government established a new bank, called the Rentenbank, in November, 1923. With that bank came a new currency, the Rentenmark, defined as equal to the gold value of a pre-war mark. Described
later by Beusch [1928, p.62] as the "miracle of the rentenmark", the rentenmark was an indexation technique which meant that paper money could be expressed in terms of real purchasing power, i.e., the purchasing power of the 1914 marks. This led accountants of the day to recommend what was called the Goldmarkbilanz technique for dealing with inflation; meaningful income measurement was to be effected using figures expressed in terms of units of similar purchasing power, in this case an historical monetary unit.

Clarke and Dean's [1989] "Conjectures on the influence of the 1920s Betriebswirtschaftslehre³ on Sweeney's Stabilized Accounting" provides a particularly helpful review of accounting thought in Germany during the period 1919-1924. Their discussion indicates that there were many books and articles about the problem, the most influential being those by Schmalenbach [1921], Mahlberg [1923a and b], and Schmidt [1924]⁴. The fact that (a) the government was forced to legislate to deal with the problem of changing purchasing power, and (b) so many accountants published books and articles on inflation accounting around that time, is consistent with the hypothesis that the high inflation of the period stimulated widespread interest in inflation accounting.

³ Business Economics theorists
⁴ These were all cited by Sweeney [1936].
Most of the normative accounting research from 1955 to 1970 was concerned with asset valuation and income determination. Literally hundreds of different authors were involved. It is difficult to argue that this was all in response to the inflation of the 1940s and 50s, but many of the major works refer to the inflation of that period as the reason for their interest in income measurement.

In the US, the year of highest inflation this century was 1946 when the CPI rose to 18%\(^5\). The following year, on 27 May, 1947, the American Institute of Accountants (later to become the AICPA) wrote to the Rockefeller Foundation requesting a grant to set up a Study Group on Business Income. The Study Group commissioned five monographs including the excellent "Income Measurement in a Dynamic Economy", by Sidney Alexander [1950] (reproduced with revisions by Solomons in Baxter and Davidson's 2nd and 3rd editions [1962 and 1977]) and stimulated much public discussion of the income measurement problem. On page 3 of its Report [1952], the Study Group quickly makes clear the motivation for its efforts:

The relation of changes in price levels to the determination of income is manifestly of crucial importance. In Exhibit II is presented a chart, made available by courtesy of Dun & Bradstreet, Inc., showing the changes in price levels and in purchasing power during the century ended 1950. The changes in prices that have taken place in the last twenty-five years have

\(^5\) Sharpe [1985], Table 1.1, shows the increase in the CPI in the US was 18.2% in 1946, and 9.0% in 1947.
emphasized the importance of this phase of the problem which the Group undertook to study, and the present report is, therefore, largely devoted to it. [1952, p.3]

*** Figure 6.1 goes approximately here, see next page ***

The Study Group's Exhibit II is reproduced as Figure 6.1. Amidst considerable dissension, the Study Group's Report [1952] recommended the use of CPP accounting to

facilitate the determination of income measured in units of approximately equal purchasing power, and to provide such information wherever it is practicable to do so as part of the material upon which the independent accountant expresses his opinion. [Para. 12, p.105]

Not long after, Jones [1955, 1956] provided empirical evidence of the measurement "error" associated with using historical cost reports in times of inflation. His first survey [1956] was of nine US steel companies representing 80% of the industry, from 1941-47. He found that although the firms' income calculated by conventional means had exceeded dividends by a substantial margin each year, after adjusting for general price-level changes real dividends had not been earned since 1941. The seven-year difference was $US750 million. The second study [1955], commissioned by the AAA, was of four non-steel companies. Again, with figures restated for inflation (general prices), Jones found gross income for 1940-51 fell from 187% to 55%, that average income tax was 67% not 57%, and that dividends exceeded real income.
EXHIBIT II

PRICES AND WAGES—1850–1950  (Wholesale Commodity Index; Wages per Hour Correlated to Same Base Year, 1926 = 100)
Prepared by Dun & Bradstreet, Inc.

SOURCES: All of the price indexes in this chart were compiled by the Bureau of Labor Statistics. They express the fluctuations in prices in terms of the average level prevailing in 1926. The Food Index includes cereal products, dairy products, fruits and vegetables, meats, poultry, and fish. The Textiles Index represents the prices of clothing, cotton goods, hosiery and underwear, rayon, silk, woolen, and worsted goods. Building Materials include brick and tile, cement, lumber, paint, and paint materials. The per hour Wages Index (1926 = 100) was constructed from dollar figures compiled by the Bureau of Labor Statistics. The data for Purchasing Power of the Dollar were computed by the Office of Business Economics from the Bureau of Labor Statistics' Wholesale Price Index (1926 = 100); the reciprocal of the Wholesale Index is converted to a relative with the 1935–1939 average as a base. The editorial comments attached to the original chart are not reproduced.
In the UK, the response to post-WW2 inflation has been described by Tweedie and Whittington [1984] as follows:

In the three years immediately following the Second World War, the UK, like the US, experienced one of the sharpest inflations within living memory (as it was at that time). This naturally led to agitation from business and from others, such as academics, for reconsideration of the traditional historical cost methods of accounting. [1984, p.44]

Responding to calls for reforms, the ICAEW issued its Recommendations of Accounting Principles, N12, "Rising Price Levels in Relation to Accounts", in January 1949. Recommendation N12 rejected any form of inflation accounting. From Tweedie and Whittington's account [1984, pp.44-46] there followed in rapid succession: (i) a Committee on the Taxation of Trading Profits (Millard Tucker Committee, appointed 1949); (ii) reports by the Institute of Cost and Works Accountants (1952) and the Association of Certified and Corporate Accountants (1952) -- both of whom proposed replacement cost valuation systems, (iii) a response to those proposals from the ICAEW Recommendations of Accounting Principles, N15, "Accounting in Relation to Changes in the Purchasing Power of Money" (1952), which reiterated its belief that historical costs should continue to be used as the basis for the accounts; and (iv) a Royal Commission on the Taxation of Profits and Income, which reported in 1955. From 1955 onwards, inflation dropped, and with it, interest in inflation accounting. In Tweedie and Whittington's words:

Inflation continued at a modest rate through the remainder of the 1950s and most of the 1960s. The rates of inflation experienced during this period (ending 1968) were at the lower end of what has been described as "the discussion range" (Burton, 1975), i.e., they were sufficient to stimulate discussion of the problem by academics and thoughtful
practitioners, but not significantly high to create urgent demands for reform from companies or users of financial statements. [1984, p.46]

In Australia, Mathews and Grant [1957] made a similar study to Jones [1955, 1956] and concluded that firms should use replacement-cost pricing and current cost accounting for income measurement. Further, they recommended that company taxes should be based on "current income, not accounting income". Their opening paragraph [p.1] explains the motivation for their study:

The last twenty years have been marked by one of the greatest inflations that the world has ever known. Although price instability is not a new phenomenon, the inflation that began with the 1939-45 War has been unique, not only in its pervasion and severity, but also because its persistence has led to the belief that prices will never recede to anything like their pre-war level.

In the fifteen years that followed, accounting research was almost synonymous with price change accounting research. Notable publications, including Edwards and Bell [1961], ARS6 (written by staff of the AICPA’s Accounting Research Division) [AICPA, 1963], the American Accounting Association’s ASOBAT report [1966], Chambers [1966], Gynther [1966], Solomons [1966], Ijiri [1967], Thomas [1969], and Sterling [1970], were summarized in Chapter 2. It would be overstating the case to say that all these studies were motivated by the inflation of the 1940s and 50s. However, there were sufficient studies citing Post-WW2 inflation as their motivation[6] to support the hypothesis that the high inflation of the period stimulated widespread interest in inflation accounting.

---

6 For example, Edwards and Bell [1961, p.13], ARS6 [AICPA, 1963, p.6-8], and Gynther [1966, p.8].
Case 3: Hyperinflation in South America

Inflation in South America during the 1970s is another obvious place to look for problems with historical-cost accounting. Referring to a delightfully absurd interaction of historical-cost thinking and foreign-currency translation, Wolk, Francis, and Tearney [1984] describe the "disappearing asset problem":

For example, assume an Argentine subsidiary purchased a fixed asset in December, 1974, when the Argentina peso-US dollar exchange rate was $0.20. The asset cost 20,000,000 pesos and would be translated as $4,000,000. By September, 1982, the exchange rate was .000040; thus, the asset would be translated at $800. [p.561]

Wolk et al. go on to explain that the FASB considered indexing the original cost of the asset, in this case in pesos, for the change in the general price level in Argentina, but this solution was rejected on the grounds that it departed from historical cost. Eventually SFAS 52 [FASB, 1981] allowed the historical US dollar cost to be reported (in place of the functional-currency cost converted through the current exchange rate) in countries where the cumulative rate of inflation in the functional currency exceeded 100% over a three year period.

Table 6.1: Percent Increase in Annual CPIs for Brazil, Year 19xx

<table>
<thead>
<tr>
<th>Year</th>
<th>70</th>
<th>71</th>
<th>72</th>
<th>73</th>
<th>74</th>
<th>75</th>
<th>76</th>
<th>77</th>
<th>78</th>
<th>79</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPI</td>
<td>23%</td>
<td>20%</td>
<td>17%</td>
<td>13%</td>
<td>25%</td>
<td>29%</td>
<td>42%</td>
<td>44%</td>
<td>41%</td>
<td>77%</td>
</tr>
</tbody>
</table>

(Source: Seed [1981], Exhibit 2-1, p.18)
Inflation at rates of 100% to 1,000% p.a. in countries like Argentina and Chile, is obviously so high\(^7\) that, as in Germany in the 1920s, accounting systems are forced to recognize its impact. More interesting is Brazil, which experienced "double digit" inflation (as measured by the CPI) in 19 of the 20 years, 1958-1977 [Tweedie and Whittington, Table 9.4, p.235], and where historical-cost accounting has also been abandoned. Chapter 10 in Seed [1981, pp.131-146] examines the management accounting practices of Acos\(^8\) Villares, S.A., Sao Paulo, Brazil, described as the largest specialty steel producer in Brazil, with revenues of USD100 million in 1979. As shown in Table 6.1, general inflation in Brazil in 1979, the year of the study, jumped suddenly to 77% per annum. It might be thought that accounting in Argentina is not relevant to, say, the US or UK, but this is not so. The techniques Seed reports being used by Acos Villares for management accounting include:

- standard cost accounting,
- direct costing,
- flexible budgeting,
- responsibility reporting,
- strategic planning,
- return-on investment analysis (partially based on current costs),
- funds flow management, and
- the evaluation of prospective capital expenditures based on their internal rate of return [p.131].

It is clear from Seed's chapter that Acos Villares is a sophisticated user of accounting information, and does much of its planning in constant cruzeiros. Departmental heads use constant cruzeiros to facilitate month-to-month and year-to-year comparisons, and prices are based on replacement costs (within the

---

\(^7\) Tweedie and Whittington [1984, p.235] report 444% inflation in Argentina in 1976, and a change in the general price index from 100 in 1970 to 253,000 by June 1980. Chile's index over the same period rose from 100 to 414,000, Brazil's from 100 to 1,030.

\(^8\) "Acos" is the Portuguese word for "steel".
constraints of government price controls) with all prices quoted as subject to adjustment based on CIP\textsuperscript{9} index changes. This evidence from Brazil is particularly interesting because it suggests that even for countries with "only" double-digit inflation, historical-cost accounting creates serious difficulties. Thus accounting practices during the South American inflation of the 1970s are also consistent with the hypothesis that the high inflation of the period stimulated widespread interest in inflation accounting. In Argentina's case that interest was sufficient to prompt government legislation.

Case 4: The Practitioners' Response to US and UK Inflation in the mid-to-late 1970s

The final case of inflation to be investigated is that in the US and UK in the mid-to-late 1970s (following the oil crisis of 1973). Reading the literature written towards the end of the 1960s, one gets the sense, e.g., from practitioners' comments at the Symposium on Asset Valuation and Income Determination organized by Sterling [1971], that practitioners acknowledged the deficiencies in historic cost accounting and were ready to accept some sort of general price-level adjusted or replacement-cost accounting system\textsuperscript{10}. Thus, when the next period of high inflation came along, they seemed predisposed to act quickly.

\textsuperscript{9} "CIP" stands for Inter-ministerial Price Control, the latter index being based on a sample of 36 of the company's products, and is computed in accordance with government regulations.

\textsuperscript{10} They would not, however, "have a bar" of net present value accounting [Staubus, 1971] or CoCoA [Chambers, 1971].
Table 6.2: Percent Increase in Annual CPIs for the US and UK

<table>
<thead>
<tr>
<th>Year, 19..</th>
<th>71</th>
<th>72</th>
<th>73</th>
<th>74</th>
<th>75</th>
<th>76</th>
<th>77</th>
<th>78</th>
<th>79</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>4%</td>
<td>3%</td>
<td>6%</td>
<td>11%</td>
<td>9%</td>
<td>6%</td>
<td>7%</td>
<td>8%</td>
<td>11%</td>
<td>14%</td>
</tr>
<tr>
<td>UK</td>
<td>10%</td>
<td>7%</td>
<td>9%</td>
<td>16%</td>
<td>24%</td>
<td>16%</td>
<td>16%</td>
<td>8%</td>
<td>13%</td>
<td>18%</td>
</tr>
</tbody>
</table>

(Sources: Seed [1981], Exhibit 2-1, p.16; Tweedie and Whittington [1984], Table 9.1, p.218)

The annual percentage increase in the CPI for the US and the UK from 1974 through 1980 is shown in Table 6.2. Whether because of this inflation, or as a flow-on from the intense academic debate of 1955-1970 (which itself was motivated initially by inflation), these relatively low levels of general inflation were enough to trigger US standards ASR 190 [SEC, 1976] and SFAS 33 [FASB, 1979]. In the UK, inflation had risen to 16% p.a. before the Sandilands Committee was established in Autumn, 1974. However, the pressure for action cannot have been too strong, because it took nearly five years of "due process" from the time the Sandilands report was published until the eventual standard, SSAP 16, was issued in 1980. One might hazard a guess that the critical level of inflation is around 20% p.a. in the UK, and possibly lower in the US. But such guesses are unimportant. What is important, however, is that when the level of inflation rose high enough, the clamour for improvements to historical-cost accounting were loud enough for the professional accounting bodies to act. Again, this is consistent with the hypothesis that the high inflation of the period stimulated widespread interest in inflation accounting.
Conclusions on the need for Inflation Accounting

The empirical evidence presented above covers most of the major experiences with inflation in the Western world this century. In each case, as the level of inflation rose there were widespread calls for improvements to historical-cost accounting. The German inflationary experience is a striking example of the failure of historical-cost accounting and the remedy was simple: use of a government-backed general-purchasing-power index. For the same reasons, government-mandated general-price-level-index adjustments are used in Brazil and Chile today [Tweedie and Whittington, 1984, pp.235-241]. The US, UK, Canada, New Zealand, and Australia have never experienced hyper-inflation, but whenever inflation has exceeded, say, 10% to 15%, for more than a few years, there have been widespread calls for improved methods of accounting for changing prices.

On the other hand, the evidence from the empirical studies reported in Chapter 3 indicates quite strongly that stock market analysts in the US and UK in the early 1980's did not treat CCA income as more relevant than GAAP income. It also suggests that analysts can make reasonable estimates of CPP

---

11 Not covered above were inflation in France in the early 1920s, inflation in Canada, New Zealand, and Australia in the 1970s, and inflation in Israel. Response to the French inflation was similar to that in Germany; response to Canadian, New Zealand and Australian inflation was similar to that in the UK; and as noted in an earlier footnote, it is hard to obtain data on Israeli inflation accounting.

12 More formally, the null hypothesis that there was no association between inflation and calls for reform was rejected in each case.

13 First in the 1940s and 50s and again in the mid-1970s.
and even CCA income by combining information from other sources with GAAP income.

How is one to reconcile the apparently contradictory evidence that inflation accounting is both necessary, and unnecessary? The reconciliation offered here is simply that inflation accounting becomes increasingly necessary as the level of inflation rises. It seems that at levels of inflation like those experienced in the US from 1980-1984, e.g., around 10% per annum, there may be some glaring cases where Historical Cost accounting produces nonsense numbers (because not all prices move in line with the general level of prices), but that these problems are still not large enough, or general enough, to be detectable in market-wide studies like those reviewed in Chapter 3. At higher levels of inflation, such as those discussed in this chapter, Historical Cost accounting breaks down, and some form of inflation accounting becomes absolutely necessary.

Now, the stated objective of this thesis (page 1-3) is to see whether cost-effective, computer-based accounting systems can be used to generate better accounting information than existing transaction processing systems. It is clear from the discussion above that at high levels of inflation, inflation accounting information is better than Historical Cost information (because HC produces nonsense), and that at low levels of inflation the information content of both systems of accounting is more-or-less equivalent. Since no one knows when the next bout of high inflation will occur, it seems wise to plan now for that eventuality, and build inflation-tolerance into everyday accounting systems. Furthermore, it seems likely that even when general price level changes are low, there may still be times when major assets or liabilities of some firms
experience large changes in specific prices, and that it would be desirable if the accounting system reported the effect of those changes. This provides a further reason for building price-change-tolerant accounting systems.

Hence Design Principle 1: It is worth trying to build a cost-competitive, general-purpose, computer-based accounting system that is able to generate reports in inflation-adjusted/current-value terms.

The reason for recommending inflation/current-value accounting, is that just as ships should be designed to cope with bad weather, so accounting systems should be designed to cope with times of high inflation. The challenge is to design an inflation-accounting system that can compete with HC in terms of simplicity, cost, reliability, auditability, and so on. The system does not have to be as simple as HC, because it potentially offers better information than an HC system.

6.1.2 A Possible Practical System of Inflation Accounting

Without in any way limiting the generality of Design Principle 1, or the system architecture to be developed in what follows, a concrete example of the sort of adjustments considered appropriate may be helpful. Given the costs of collection and the uncertainties and difficulties of estimating and verifying

---

14 The arguments supporting market efficiency suggest that managers and investors would learn about such major price changes well before they were published in accounting reports. However, this is not an argument for excluding the information from those reports.
replacement costs and/or current cash equivalents, a reasonable, low-cost form of inflation accounting might be as follows:

1. Subject to conformity to trends in specific price index series\textsuperscript{15}, use the lower of latest purchase cost and market prices for inventory valuation.

2. As suggested by Carsberg [1984a], use a small number of government-certified price index series to restate fixed asset costs to rough-and-ready current-values (with specific revaluations encouraged for major items).

3. Use conventional methods of depreciation.

4. Use financial capital maintenance for income determination, including recognizing gains and losses on net monetary items.

Such a system would be more robust in the face of inflation than historic cost accounting, and not much more expensive to run. So much that is important to estimating future cash flows is ignored in any accounting system that any further efforts towards "accuracy" are probably unwarranted.

\textsuperscript{15} This proviso is to prevent deliberate manipulation of inventory values by buying in small lots or at artificially high prices.
6.2 Design Principle 2: The Corporate Database

Since data must be collected for the firm's special-purpose feeder systems for day-to-day control purposes, they should be retained in machine-readable form as a corporate database for subsequent analyses.

Design Principle 2, stated above, is a conclusion drawn after considering the implications of the findings reported in Chapters 4 and 5 for the design of future computer-based accounting systems. It is far less contentious than Design Principle 1, and is therefore simpler to justify.

6.2.1 Justification of Design Principle 2

Chapter 4, it will be recalled, examined computer-based accounting as it is practiced today. Present-day TPAS were found to be structured more-or-less as shown in Figure 4.1, with most data being collected in a series of special-purpose subsystems that have been carefully tailored to the needs of the host organization. From these systems, summary journal entries were extracted and retrieved into the firm's general ledger. Chapter 4 concluded that developments in computer-based accounting systems have been aimed at assisting management by making day-to-day recording and reporting easier to do. As a result, computer-based TPAS are often powerful, sophisticated, comprehensive systems, but they have not attempted to improve or change the underlying system of ledger accounts with debits and credits of constant money amounts that has prevailed in accounting, conceptually unchanged, for literally hundreds of years.
Chapter 5 examined the academic literature on what was loosely termed "computer-based accounting". The two most important contributions identified were Ijiri's Multidimensional Accounting [1975] and McCarthy's REA metamodel [1982]. Ijiri's contribution was to recognize that the money amounts debited and credited by accountants can always be viewed as a quantity of some resource multiplied by a money value for each unit resource. Equally important is Ijiri's notion of causal double entry: the value assigned to the resources obtained in an exchange is usually determined by the value of the items given up in the exchange. McCarthy appears to have been the first accountant to suggest that conventional double-entry accounting should be regarded as an external schema, with data derived from the firm's underlying database of facts about resources, events, and agents. His REA model may be viewed as a development of both Sorter's Events Accounting [1970] and Multidimensional Accounting, though McCarthy explicitly sought to reject using the notion of an account as a modelling concept for the underlying database.

Design Principle 2 is an attempt to link these two apparently independent streams of thought; it relates the computer systems described in Chapter 4 to both Sorter's Events Accounting and McCarthy's REA model discussed in Chapter 5. A problem for all those who have considered the merits of Events Accounting has always been to decide what facts should be captured and stored in the database. However, in the 20 years since Sorter's paper, computers have grown more and more powerful, and less and less expensive. Computer-based information systems have grown with them. It is clear from Chapter 4 that a multitude of facts is presently being fed into computer-based TPAS every day. It also seems likely that the trend towards more and more complex systems will
continue. Why should it stop? If more sophisticated data models can be built and maintained at reasonable cost, management will presumably be better off. The choices about what facts should be stored in these special-purpose systems are dictated by the needs of the organizations that use them. Already, systems like Weber's [1986] wholesale distribution software (discussed in Chapter 5) are much more complex, and contain far more detail, than was ever possible in any accounting system prior to the advent of computers.

Stepping forward to, say, the year 2000, it seems reasonable to predict that much of the data required for accounting systems at the general ledger level will be available from the special-purpose feeder systems. Of course, some data will not. For instance, estimates of replacement costs, asset lives, net realizable values, and differential cash flows for DCF calculations, will still have to be specially collected when and if required. However, many standard accounting reports will probably be generated directly from data extracted from the feeder systems. In a sense, time, and the needs of modern organizations for information, will have rushed past Events Accounting. Accountants in individual businesses will not have to decide what information should be recorded in each event record; it will have been decided for them by software designers. In many cases it will be a much richer picture of the activities of the firm than Sorter [1969] would have imagined possible.
Now, in systems with such sophisticated models of reality it seems most unlikely that designers will omit key economic variables that record the R’s, E’s and A’s of McCarthy’s REA model. For instance, consider the sales orders investigated by Weber [1986]. Assuming that sales orders relate to the future delivery of goods to different customers, the absolutely essential characteristics of ANY sales order are details of the goods ordered, some identifier of the sales order (e.g., date, sales order number), and the identity of the customer. These are shown in Figure 6.2 with the R’s, E’s, and A’s from McCarthy’s model highlighted in bold print. Therefore, even before reading Weber’s [1986] paper one could safely predict that all order entry packages would describe the resources ordered (McCarthy’s R), details of the order event such as the date, sales order number, purchase order number, and so on (McCarthy’s E), and the customer (McCarthy’s A). That is what order-entry systems do. Some systems do much more; none do less. In other words, with the exception of McCarthy’s duality relationship (discussed in Chapter 5), it should ALWAYS be possible (trivially so) to derive an REA view from an order-entry database. Weber’s paper may be viewed as confirmation of this prediction.
It is therefore predicted that as more and more sophisticated special-purpose computer information systems are built, primarily for short-term control purposes, NONE will omit the key relevant economic variables, i.e., the R’s, E’s, and A’s of McCarthy’s model. The systems would simply be unworkable if these key economic variables were not captured. Furthermore, because these systems must capture these R’s, E’s, and A’s, their databases will provide an excellent source of data for income measurement. Both Sorter’s and McCarthy’s goals for Events Accounting will be met if accounting systems can scan through the facts stored by these special-purpose systems, extracting and reorganizing data for summary accounting reports.

Design Principle 2 follows: Since data must be collected for the firm’s special-purpose feeder systems for day-to-day control purposes, they should be retained in machine-readable form as a corporate database for subsequent analyses.

For most firms, detailed data describing almost all relevant economic events are or will be available in machine readable form as they are processed through the relevant subsystems. The only additional costs associated with Design Principle 2 are for organizing and storing data until they are discarded. As storage costs continue to fall, it seems likely that firms will find it useful to store all transaction information for quite long times, e.g., three to five years, possibly more. If storage costs prove excessive, it should still be possible to retain meaningful aggregations of more detailed data for analysis purposes.
6.2.2 Summary: Design Principle 2 (the corporate database)

As more and more special-purpose systems are developed, primarily for day-to-day control and planning purposes, more and more facts about the firm will be available in machine-readable form at low cost. There is little point trying to predict what additional data firms will store, and different industries may have quite different needs. But their common need is for income and wealth measures, and it may safely be predicted that almost all the transaction data (the R's, E's, and A's) for these measures will be captured by each firm's special-purpose subsystems. The prime recommendation presented in this section, Section 6.2, is that the facts from all these data-rich subsystems should be saved in some sort of corporate database for analysis purposes, including the generation of inflation-tolerant income and wealth measures. It seems reasonable to predict that future accountants will spend less time preparing routine reports, and more time making special-purpose analyses of data extracted from these corporate databases.

6.3 Consistency of the above two Design Principles with the Objectives of Accounting identified in Chapter 2

It would be unwise to proceed without checking to ensure that Design Principles 1 and 2 are consistent with the three Objectives of Accounting identified in Chapter 2. Such checks are rather mechanical, so they have been relegated to Appendix 6.1. Briefly, neither Design Principle is inconsistent with the three Objectives. The benefits of Design Principle 1 will only become apparent in times of high inflation, or more generally, large price changes. During such
times, Design Principle 1 assists with all three objectives (i.e., for management information, earning power predictions, and accountability reporting). The major potential benefits of Design Principle 2 are for management analyses, and as a source of data for the generation of inflation-tolerant financial accounting reports (i.e., as support for Design Principle 1).

6.4 An Architecture for Computer-based AIS

This section marks the turning point in this thesis. The important accounting decisions about the goals for computer-based accounting systems have been made. The task now is to transform the two general Design Principles from Section 6.1 and 6.2 into an architecture for the design of future computer-based accounting systems. From this point on, the thesis becomes increasingly technical.

The first Principle from Section 6.1 requires that future accounting systems be inflation-tolerant; they should not generate misleading information in times of inflation. The second Principle requires that whenever new computer systems are being developed for day-to-day management and control, accountants should ensure that sufficient data may be extracted from these systems for inflation-tolerant income and wealth measures to be generated. It is considered highly likely that these special-purpose systems will contain, or have electronic access to, almost all relevant facts necessary for income and wealth measurement.

The design problem now, portrayed in Figure 6.3(a), is to take the databases of facts captured and recorded by the special-purpose subsystems and to use them
to assist with the generation of current-value accounting reports. How can this be done in as general a way as possible?

*** Figure 6.3 goes approximately here, see next page ***

As an anchor point, Figure 6.3(b), based on Figure 4.1, shows the method used in computer-based accounting systems at present. General Ledger journal entries are either generated by, or extracted from, the various subsystems, then fed to a general ledger package. Note that the interface programs shown in Figure 6.3(b) may be quite simple. For instance, journal entries from Accounts Payable, Accounts Receivable, and Inventory Control in the ACCPAC system reviewed in Section 4.2 are generated automatically by those systems, for retrieval directly into ACCPAC's GL system, without conversion.

There are two weaknesses with the system outlined in Figure 6.3(b). First, it may be difficult to use the data in the subsystem databases effectively. Even when interface programs exist the information they provide is likely to be very restricted. One has to rely on all the different subsystems and interface programs generating appropriate journal entries, and changes in accounting policy, even simple changes like a revision to the chart of accounts for the general ledger, will have implications for many different subsystems. Second, conventional general ledger systems are decidedly intolerant of inflation. The balance in a general ledger account is constant, in money terms, until it is changed by a journal entry. It is then constant again, until the next journal entry. However, market values are constantly changing, particularly in times of inflation, so accounting information may be more useful if values of assets and
Databases from:

subsystem 1 ———

subsystem 2 ———
   : 
subsystem n ———

special information
   e.g., replacement costs ———

Inflation-Adjusted Accounting reports

Figure 6.3(a) The design problem

Databases from:

subsystem 1 —→ interface 1 ———

subsystem 2 —→ interface 2 ——— J/E
   : 
subsystem n —→ interface n ———

journal entries keyed-in ———

GAAP Accounting ledger reports

Figure 6.3(b) The present solution

Databases from:

subsystem 1 —→

subsystem 2 —→
   : REE
   : data —→ Interpreter
subsystem n —→

model ———

special information
   e.g., replacement costs ——

price index series ———

Formula FAJE Accounting
   General Ledger

Inflation-Adjusted Accounting reports

Figure 6.3(c) Proposed Architecture for Computer-based Accounting

G-25(a)
liabilities, and even balances in revenue and expense accounts are continuously restated.

To overcome these two quite independent problems it is proposed that future computer-based accounting systems should be designed as shown in Figure 6.3(c). Figure 6.3(c) is the proposed architecture for computer-based accounting systems, the major research contribution of this thesis. The novel features of Figure 6.3(c) are first, Formula Accounting, and second, the Interpreter combined with the Resources and Exchange Events (REE) model. The Interpreter and Formula Accounting could be implemented quite independently of one another, but both have the potential to make significant improvements to the present state of computer-based accounting.

To emphasize this independence, there are two paths from the subsystem databases in Figure 6.3(c) to the Formula Accounting general ledger. In the upper path it is assumed that interface programs similar to those used in Figure 6.3(b) generate the FA journal entries. In the lower path, a database schema called the REE model and a program called an Interpreter have been interposed between the subsystem databases and the FA general ledger. Because Formula Accounting is common to both paths it will be discussed first.

6.4.1 Formula Accounting

In a Formula Accounting system, the journal entries and ledger balances contain formulae, not just constants. Evaluation of a formula occurs whenever an account is used for reporting, or is displayed on the screen. This is exactly
analogous to the way formulae in spreadsheets are displayed as values when viewed on the screen. In Figure 6.3(c) a Formula Accounting general ledger replaces the conventional general ledger from Figure 6.3(b). Data flows from the subsystems to the Formula Accounting general ledger have also been transformed from conventional journal entries in Figure 6.3(b) into Formula Accounting journal entries (FAJEs) in Figure 6.3(c).

Formula Accounting (FA) is primarily intended to solve the problems of using inflation accounting in a conventional general ledger framework, but it is also useful for HC accounting. The examples in the next chapter, Chapter 7, illustrate both HC/GAAP and CC/CPP accounting journal entries. An implementation of Formula Accounting, which amplifies the examples in Chapter 7, is presented in Chapter 8.

6.4.2 The Interpreter

If an organization chose to generate its Formula Accounting journal entries using the lower path of the Figure 6.3(c) architecture it would need an REE data model and an Interpreter. As envisaged in the proposed architecture, the Interpreter's job would be to make important accounting decisions like those made by a human accountant in today's systems, e.g., how assets should be valued, and how the real world is to be modelled in terms of journal entries. Because of the importance of valuation and interpretation of accounting rules in income measurement, there may be advantages if all these accounting decisions are made in a separate module, quite distinct from any of the subsystems feeding it with information. Thus, in the lower path proposal, the Interpreter,
not the interface programs, contains the valuation and accounting rules required to generate the FA journal entries.

The two main reasons in favour of a separate Interpreter approach are:

(a) Modular, loosely coupled systems, are usually more robust and error-free than tightly-coupled systems. Presumably specialist subsystems will be rewritten, replaced, split in half, integrated, and added to, at different times. In addition, the accounting system itself will be probably need to change due to changing accounting standards. In this ever-changing world it may be advantageous to concentrate responsibility for accounting rules, including knowledge of the general ledger chart of accounts, in one module, not to spread that responsibility over the various subsystems.

(b) Not all valuation information is available in the subsystems anyway. Data on current replacement costs and net realizable values (if required) has to be collected externally. This is indicated by the "special information" feeding up into the Interpreter in Figure 6.3(c). Again, this points to the possible merit of centralizing generation of Formula Accounting journal entries.

Details of a prototype implementation of an Interpreter are given in Chapter 10.
Potentially the greatest difficulty with the lower path proposal in Figure 6.3(c) is the definition of the corporate database that provides data for the Interpreter. All that the Interpreter "sees" when it looks for data is a database in REE format. Data of the type listed in Table 2.1 must be collected from the various interface programs.

Implicit in the lower path in Figure 6.3(c) are a number of new interface programs that provide the Interpreter with less-processed information about economic events than the journal entries from interface programs 1, 2, etc. in Figure 6.3(b). The challenge is to find a data model that is as general as conventional double-entry accounting, yet with a sufficiently rich description of the economic resources and events that have occurred for inflation-adjusted income and wealth measures to be generated automatically. Whether this corporate database is implemented as a physical or virtual database is unimportant. Figure 6.3(c) is drawn for the case where it is a virtual database, a view derived from the subsystem databases. However, if the subsystem databases were automatically purged of old transaction data, the relevant information would have to be saved in a real database.

The most obvious candidate to turn to for a general data structure for the corporate database is McCarthy's [1982] REA model, previously discussed in Chapter 5. Could a series of REA views derived from the subsystem databases be used as a sufficiently complete, and general, source of data for the Interpreter? The answer is that it does seem that something similar to a
Resource-Events-Agents framework would be suitable for the detailed historical record requirements of types (i) and (ii) from Table 2.1. For instance, Ijiri's Multi-dimensional Accounting [1975] is based around the assumption that details of a firm's sales and purchases (including purchases of services like labour, advertising, and rent), borrowings, lendings, and capital transactions, can all be thought of as events where resources are exchanged. This can be extended to a manufacturing firm, by viewing the various inputs being exchanged for the outputs of each production process.

However, as discussed in Chapter 5, there are some problems with using REA as the interface model. For this reason a new data model, similar in scope to McCarthy's REA, was developed for use with the prototype Interpreter. That model, called the Resources and Exchange Events model (REE) is presented in detail in Chapter 9. The REE model refines and extends REA in the two following areas:

(a) REE uses "accounts" as a modelling primitive. As discussed in Chapter 5, McCarthy was adamant that REA-type models should not include "elements of double-entry bookkeeping such as debits, credits and accounts" [1982, pp.559-560]. He therefore developed his causal chain concept of duality in an attempt to link events like sales and cash receipts, and so avoid concepts like accounts receivable. However, when Ijiri's causal double-entry approach is used as the theoretical basis for asset valuation on date of exchange, an account-based approach, like his Multi-dimensional accounting, is actually quite a useful way of describing relevant economic events. In addition, an account-based approach makes it relatively simple
to extract resource-event information from subsystems like Accounts Payable and Accounts Receivable that use an explicit account-based framework for recording economic events. This is an important consideration because the Accounts Payable subsystem, in particular, is the source of many journal entries in most present-day accounting systems.

(b) Resource description is essential for the Interpreter. McCarthy's REA paper is almost silent about how resources change value over time. In his Figure 10 [p.574] he shows Depreciation, and Advertising Service Consumption, as two examples of economic events, in an awkward construction he calls "macro-level duality". The position taken in this thesis is that changes in value over time are not economic events (the only "events" modelled in REE are "exchange events"). Instead, all resources are assumed to change in value over time, and resource-value modelling in REE enables the Interpreter to meet the asset valuation requirements of Table 2.1. All resources in the REE model are potentially linkable to a price index series of some kind. In the case of depreciable-type assets, the resource description also includes information about the life of the asset and the expected pattern of future benefits.

6.4.4 Multiview Accounting

The final observation to be made about the lower path of the architecture in Figure 6.3(c) is its potential for generating alternative views of economic reality. If the special-purpose subsystems continue to evolve as predicted, almost all journal entries posted to the Formula Accounting general ledger will
be generated automatically from data extracted from them. It would therefore be relatively painless to maintain a number of different general ledgers, each using a different set of accounting rules. A system with more than one set of "books" may be called a "Multiview Accounting system" since it provides multiple accounting views of the underlying economic reality.

If current values for assets, liabilities, revenues, and expenses, were used in the firm's normal books, but not allowed for taxation purposes, there would be a real need for two full sets of books. Figure 6.3(c) could therefore be modified to include two separate Interpreters and two general ledgers (or one Interpreter programmed to feed two separate general ledgers). One ledger would be for historical cost tax accounting, the other for current-value accounting. Of course, many organizations today say they keep two sets of books, the normal books, and the tax books, but this is usually just a matter of duplicate records with different depreciation rates in the Fixed Asset system. Unless the taxation law is changed, any move to, say, CC/CPP accounting for normal reporting purposes would make Multiview Accounting essential.

Aside: In present-day computer-based accounting systems, like those portrayed in Figure 6.3(b) (not (c)), the idea of maintaining two full sets of books would probably not be feasible. It takes so much highly skilled management effort to ensure that one system works that proposals for a second system would be resisted strongly. It is this need for constant managerial input that makes present-day accounting systems so difficult to use. Yet as systems become more automated, and more and more routine accounting is taken over by
programs like Interpreters, it becomes quite feasible to consider multiple sets of accounting "books".

Multiview Accounting is discussed in more depth in Chapter 10, the chapter on the Interpreter.

6.5 Chapter Summary

This chapter is both the culmination of Chapters 2 through 5, and the foundation for the remainder of the thesis. The decisions made in this chapter determine the direction for the remainder of the thesis. In Sections 6.1 and 6.2, after careful consideration of the implications of Chapters 2 through 5, two Design Principles for the design of future computer-based accounting were enunciated and justified. In Section 6.4, these Principles were represented graphically in Figure 6.3, and the requirements for the computer-based accounting system of the future began to emerge. Figure 6.3(c) shows the outcome of the design choices considered in this chapter. It is the proposed "Architecture for Computer-based Accounting Systems" that motivated the title for this thesis.

There are actually two alternative system designs within the one architecture shown in Figure 6.3(c):

(a) In the upper path in Figure 6.3(c), the separate feeder systems generate Formula Accounting entries either themselves or via the interface programs. The upper path avoids the need for an Interpreter and data transformation into REE format.
(b) In the lower path in Figure 6.3(c), Formula Accounting journal entries are generated by the combination of the REE model and the Interpreter.

Both systems satisfy the constraints imposed by the two Design Principles, i.e., they both exploit the availability of data in the various specialized feeder systems (Design Principle 2), and they both meet the need for inflation-adjusted/current-value accounts (Design Principle 1).

The lower path appears more complex than the upper path because it requires definition of the REE data model and construction of the Interpreter. However, in the upper path Interpreter-like functions would still have to be built into the subsystems or their interface programs. (Otherwise they could not generate the appropriate Formula Accounting journal entries.) Moreover, since the Formula Accounting journal entries from either path are the same, the data requirements of both paths must be identical. For this reason, the remainder of this thesis will be devoted to an investigation of the lower path in Figure 6.3(c). This has the advantage that the data requirements, and the rules for generating FA journal entries, will be made quite explicit.

In the remainder of the thesis the components of the architecture depicted in Figure 6.3(c) are considered in more depth. Chapter 7 explains how a Formula Accounting system would work. Chapter 8 reinforces these examples by describing a prototype implementation of Formula Accounting in a programming language called Prolog. Chapter 9 introduces the Resource and Exchange Events model. This model is much more complex than the simple debits-and-credits model used in conventional accounting systems. It has to be complex.
because it is attempting to store sufficient information about the real world for the Interpreter to replace most of the labour-intensive human judgement that goes into accounting systems today. Chapter 10 develops a set of rules to be encoded in the Interpreter. The Interpreter is to act like an expert system, generating Formula Accounting journal entries from data extracted from the subsystems.

(8,400 words)
Appendix 6.1: Reconciling the Two Design Principles from this Chapter with the Objectives of Accounting from Chapter 2

Table A6.1 The Objectives of Accounting (from Chapter 2)

<table>
<thead>
<tr>
<th>Objective</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>To provide management with information useful for planning and control.</td>
</tr>
<tr>
<td>2.</td>
<td>To provide external users with information useful for predicting the future cash flows and the earning power of a firm.</td>
</tr>
<tr>
<td>3.</td>
<td>To provide accountability information useful for contract monitoring.</td>
</tr>
</tbody>
</table>

Table A6.1 reproduces the three main Objectives of Accounting identified in Chapter 2. To ensure that Design Principles 1 and 2 are consistent with these Objectives, their implications for each Objective will now be considered, briefly, in turn.

Design Principle 1 and the Objectives of Accounting

(a) Design Principle 1 (Inflation accounting) and Objective 1 (To provide management with information useful for planning and control).

One of the fundamental questions a manager must ask time and time again is "Is my firm better off if I sell item X at price P or not?" To answer this question the manager must have some way of estimating the value to the firm of retaining X. At this micro level, deprival value seems the most relevant of all the valuation systems reviewed in Chapter 2. As an approximation to deprival value, it seems reasonable to request that an accounting system should be able to provide gross profit reporting based on current replacement costs (as, to some extent, reports based on LIFO inventory valuation do).

After many such micro decisions have been made, it is also reasonable for an owner or manager to ask about the overall state of affairs. How much better off is the firm as a result of this period's activities? This question, and approach, seem to be the basis for the accounting practice of matching expenses against revenue. It was shown in the cases above that in times of high inflation, Current Cost Income from Continuing Operations (CCIFCO) or even Constant Purchasing Power Income from Continuing Operations (CPPIFCO), should be a more useful performance indicator than historical cost. For example, it is apparent from the Acos Villares example in Case 3, Section 6.1.1, that when

---

16 As with all accounting systems, CCIFCO ignores changes in probable future cash flows. In a stable, mature corporation this may not be a problem. In a small growth company, for example Sarich Corp., measuring CCIFCO may be like measuring the tail of the donkey and ignoring the body. Managers would have to rely on other measures of success.
inflation was high, some form of inflation accounting became necessary for
management planning and control.

Evidently, therefore, Design Principle 1 is not inconsistent with Objective 1. Its
importance, compared with the use of Historical Cost figures, will depend on
the prevailing level of inflation.

(b) Design Principle 1 (Inflation accounting) and Objective 2 (To provide
external users with information useful for predicting the future cash flows
and the earning power of a firm).

Much of the debate in the accounting literature for the last 30 years has been
over the impact of inflation accounting on procedures for estimating future cash
flows and earning power. Obviously, therefore, it cannot be resolved here. But
in the context of very high inflation, the issue is considerably clearer. It is
clear that:

(i) Irrespective of the level of inflation, in an active securities market, market
efficiency, even with its limitations (Section 3.1), ensures that disclosure is
more important than any single income measure or asset valuation
technique [Beaver, 1981]. (For instance, one can be reasonably confident
that even if inflation were running as high as, say, 40% p.a., analysts with
access only to historic cost accounting reports, would be able to make a
reasonable job of estimating a firm's income in current terms.)

(ii) As shown by the piano retailing example in the case in Section 6.1, in times
of very high inflation, historical cost income measures are meaningless,
whereas an inflation-adjusted income measure, e.g., CPP or CCA income
from continuing operations, is likely to be much more useful as an indicator
of future earning power.

(iii) In times of very high inflation, inflation-adjusted asset values (CPP, CCA,
or CoCoA) are more relevant to almost any decision than "vanishing" HC
asset values.

However, under relatively low inflation in the US and UK in the early 1980s, the
stock markets of the world behaved as if HC earnings and CCA earnings had
very similar information content. Thus Design Principle 1 is consistent with
Objective 2. Its importance will, however, depend on the prevailing level of
inflation.

(c) Design Principle 1 (Inflation accounting) and Objective 3 (To provide
accountability information useful for contract monitoring).

Much detailed accountability information is captured and stored by feeder
systems such as accounts receivable, accounts payable, and inventory control.
Presumably such accountability information does not need to be expressed in
inflation-adjusted terms.

However, some accountability and contract monitoring goals need summary
data, and GAAP figures are often used in this role. The problem here is that
many of one's normal assumptions about the usefulness of GAAP summary figures are shattered if one imagines accounting in a time of 40% or higher annual inflation. For instance, consider the case of debt covenants. How are concepts like principal and interest defined if inflationary expectations (and interest rates) for next year are somewhere between 20% and 70%? Will the tax laws allow firms to treat interest payments at 50% p.a. as fully tax deductible expenses? If the tax laws are defined relative to an inflation-adjusted government bond, like the Brazilian ORTN, then one has inflation accounting by default. Concerning executive compensation plans, in times of high inflation it would seem rational for shareholders to prefer inflation-adjusted measures of income to historic cost income as the basis of rewarding management for its performance.

Thus Design Principle 1 is not inconsistent with the accountability goal of Objective 3, and it could positively assist with contract monitoring.

Design Principle 2 and the Objectives of Accounting

(a) Design Principle 2 (the corporate database) and Objective 1 (To provide management with information useful for planning and control).

There are two reasons for believing that a corporate database would be desirable for management accounting. First, the experience with market-based accounting research has been that when a database of facts was collected and subjected to sophisticated analysis, some most important facts emerged. This was certainly the case with the CRSP\textsuperscript{17} database at the University of Chicago in the late 1960's, and all the studies examined in Section 3.2. Design Principle 2 may therefore be assist with the discovery of hitherto unknown relationships to improve management planning and control.

Second, crude techniques such as allocating overhead on single variables, e.g., direct labour, have been strongly criticized by authors such as Johnson and Kaplan [1987] and Foster and Horngren [1988]. One of the strongest messages to come from Foster's 1988 seminar at the Melbourne Graduate School of Management was that if accountants are going to meet the challenge of new manufacturing techniques, such as Just-in-Time and Flexible Manufacturing Systems, they need to analyze data in new ways. Management accountants of the future using sophisticated analytical techniques like input-output analysis and multiple OLS regression will need data. It seems highly likely that the data they will need for such analyses are already being captured and discarded by production planning systems in many firms around the world. Evidently, therefore, Design Principle 2 is not inconsistent with Objective 1. If the corporate database facilitates data retrieval, it may even make a positive contribution towards improved information for management.

\textsuperscript{17}Centre for Research in Security Prices, sponsored by Merrill Lynch.
(b) **Design Principle 2** (the corporate database) and **Objective 2** (To provide external users with information useful for predicting the future cash flows and the earning power of a firm).

To protect themselves from unwanted scrutiny, management is unlikely to make its corporate database available to external users. However, as will be shown in subsequent chapters, such a database could be used as Sorter and McCarthy have suggested, to provide alternative accounting income and wealth measures that are not readily available from present-day systems. At present, journal entries are generated by feeder systems and fed into general ledger systems to generate a single conventional set of GAAP-style accounting reports. The emphasis in most general ledger systems today (Section 4.2) is on classifying information at the most detailed cost centre, then aggregating to any required level. Actual figures, month by month, are compared with budgets, which according to Seed [1981] are likely to include expectations of inflation. As presently structured, alternative views of those data are very difficult to generate. Of the software reviewed in Chapter 4, only the Fixed Asset package (Section 4.2.5) generated alternative figures, i.e., CCA fixed asset values.

Under Design Principle 1, future accounting systems would be required to generate inflation-adjusted figures. To do so they will need access to reliable data. It follows that since Design Principle 2 could assist with provision of that data, it could make a positive contribution towards Objective 2.

(c) **Design Principle 2** (the corporate database) and **Objective 3** (To provide accountability information useful for contract monitoring).

A full transaction history of economically significant transactions is required for accountability reporting. Such histories are already maintained today for company-law requirements, though usually on paper or microfilm. As storage costs of machine-readable media, relative to paper, continue to fall, maintaining those records in machine-readable form seems an obvious next step. The only advantage, in terms of accountability, of storing details of all transactions in some corporate database would be the possibility of generating some reports that might otherwise not be generated. In short, Design Principle 2 does little to assist with Objective 3, except indirectly through inflation-adjusted reports for contract monitoring in times of inflation. It is, however, not inconsistent with Objective 3.

---

18 Q.15 of Seed's questionnaire [1981, p.235] suggests that 58% of 282 respondents to his NAA-sponsored study of Fortune 1,000 companies used profit plans in nominal dollars adjusted for inflation. Another 23% used profit plans in constant dollars. Only 7.4% of respondents reported that inflation was "not considered in the planning process". (But Seed's response rate was only 28%, so one wonders what the proportions would have been for non-respondents!)
Part C: Details of the Proposed Architecture
Chapter 7: Formula Accounting

The purpose of this chapter is to describe the Formula Accounting component of Figure 6.3(c). Formula Accounting is a simple extension to computer-based double-entry accounting that enables continuously changing asset and liability values to be modelled as easily as discrete events. In a Formula Accounting general ledger system, constant-value journal entries and constant-value account balances are replaced by formulae like those in spreadsheets. Whenever an account balance is displayed on a computer screen or in a report, the computer simply evaluates the formula and displays the number. This is analogous to the way spreadsheet systems contain formulae in various cells, but display numbers on screens and reports.

Formula Accounting (FA) is useful wherever asset or liability values change, either directly or indirectly, with time. For instance, suppose a share trader purchased shares for $1,000 at a time when the stock-market price, p, is $2. The debit to the Shares account in FA is recorded not as 1,000, but instead as $1,000*(p/2) = 500*p. As the share price, p, changes over time, the value of the Shares account changes automatically. If p doubles to $4, the account balance displayed will be 500*4 = $2,000.

Ledger account balance formulae may depend on any well-defined independent variable(s). In the previous example, the single independent variable was the price of a share on the stock market. In other cases it can be the time since an asset or liability was acquired, a foreign currency exchange rate, the balance of
some other ledger account, a general or specific price index, the market interest rate for securities of a given risk and duration, or the number of units produced by a machine. All these independent variables change in value over time.

Section 7.1 below presents some simple examples of Formula Accounting for historical cost/generally accepted accounting principles (HC/GAAP) accounting. Section 7.1.1 uses formulae based on time; Section 7.1.2 uses formulae based on current market prices; Section 7.1.3 uses formulae based on other ledger account balances. Lotus 123 formulae are used in Section 7.1.1 to provide a concrete example of the spreadsheet formulae that might be used in Formula Accounting.

Sections 7.2 and 7.3 take Formula Accounting beyond the confines of HC/GAAP accounting. Section 7.2 uses formulae based on price indices to show how Formula Accounting may be applied to inflation/current value accounting. Section 7.2.1 presents CC/CPP equivalents of Section 7.1.1 fixed asset accounting journal entries. Section 7.2.2 shows how Formula Accounting handles CC/CPP trading account journal entries. Section 7.3 uses Formula Accounting to value a fixed-interest liability at the prevailing market rate of interest. Whilst valuation of a stream of future cash flows is conceptually simple, it is difficult to express this value in terms of a single, one-line formula. Thus the example in Section 7.3 shows how the notion of a formula may be extended to include a reference to a spreadsheet cell, or a call to a program.

Having shown how through the various examples in Section 7.1 through 7.3 that
Formula Accounting does actually work, Section 7.4 goes on to make some general comments on Formula Accounting. Section 7.5 considers how Formula Accounting could be used to modify today's conventional accounting software, i.e., in situations where there is no Interpreter to generate the Formula Accounting journal entries. The chapter is summarized in Section 7.6.

7.1 Examples of HC/GAAP Formula Accounting

7.1.1 Formulae Based on Time

*** Figure 7.1 goes approximately here, see next page ***

Assume that a fixed asset is acquired as described at the top of Figure 7.1. In HC/GAAP accounting the appropriate journal entries to record the transaction are shown in Figure 7.1 as lines (7.1.1) through (7.1.4). When these entries are posted to the FA general ledger as formulae, not money amounts, all symbols except d, the current day, will have numerical values\(^1\). Lines (7.1.1) and (7.1.2) are entirely conventional. Lines (7.1.3) and (7.1.4) would save the average large firm 12 or 13 postings of 'monthly' standing depreciation journal entries each year. If \(a = 36,500\) and \(n = 10\) years, the formulae debited and credited in lines (7.1.3) and (7.1.4) would both be \(10^* (d - d_0)\). Whenever the computer is asked to display the balance in either account, it obtains the date from its system.

\(^1\) In this and subsequent examples this chapter, subscripts 0, 1, 2, etc. indicate successive times when journal entries need to be recorded in the accounting system. In this example, journal entries are needed on the date of acquisition of the asset and at the end of each financial year. All subscripted variables throughout the chapter represent constants.
Data: On day $d_0$, a fixed asset is acquired for $a$. The asset's expected life is $n$ years and its residual value is expected to be zero. Straight-line depreciation is considered appropriate. The end of the first financial year after asset acquisition is day $d_1$, $(d_1 - d_0) \leq 365$ days. The end of the next financial year is $d_2$.

Note: In this and subsequent examples this chapter, subscripts 0, 1, 2, etc. indicate successive times when journal entries need to be recorded in the accounting system. In this example, journal entries are needed on the date of acquisition of the asset and at the end of each financial year. All subscripted variables represent constants.

### Formula Accounting Journal Entries

#### Day $d_0$:

**Dr Asset**

<table>
<thead>
<tr>
<th>Cr Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
</tr>
</tbody>
</table>

(7.1.1)

(7.1.2)

**Dr Depreciation Expense**

<table>
<thead>
<tr>
<th>Cr Accumulated Depreciation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a/(365\times n) \times (d - d_0)$</td>
</tr>
</tbody>
</table>

(7.1.3)

(7.1.4)

#### Day $d_1$:

**Dr Profit and Loss**

<table>
<thead>
<tr>
<th>Cr Depreciation Expense</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a/(365\times n) \times (d_1 - d_0)$</td>
</tr>
</tbody>
</table>

(7.1.5)

(7.1.6)

#### Day $d_2$:

**Dr Profit and Loss**

<table>
<thead>
<tr>
<th>Cr Depreciation Expense</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a/(365\times n) \times (d_2 - d_1)$</td>
</tr>
</tbody>
</table>

(7.1.7)

(7.1.8)

**Figure 7.1: Historical Cost Formula Accounting for a fixed asset**
memory, evaluates the formula, and displays the result on the screen. If \(d\) and \(d_0\) are treated as integers the balances would be constant for the whole day.

Three points about the formulae in lines (7.1.3) and (7.1.4) should be noted. First, if time were measured in months, not days, the balances would be constant for the whole month. Such details of presentation are just a matter of user preference. Second, the values of the formulae in lines (7.1.3) and (7.1.4) should not be allowed to exceed the cost of the asset. In Lotus 123 this condition could be expressed by either of the two following functions:

\[
@IF((d-d_0)\leq365*n, a/(365*n)*(d-d_0), a)
\]

\[
@MIN(a/(365*n)*(d-d_0), a)
\]

To keep the examples simple, functions like @IF and @MIN are not shown in the equations in this chapter. Third, it should be apparent that more complex formulae could be used for accelerated depreciation, for dealing with leap years, and so on. The ease with which this can be done will become more evident later in the chapter when the notion of a spreadsheet 'local' to each journal entry is introduced.

Suppose the firm closes its expense accounts annually. The first time this happens after acquiring the above asset is on day \(d_1\), where \((d_1-d_0)\leq365\). The second time is another year later, on day \(d_2\). How is this closing of expense accounts implemented in FA? The answer is shown in journal entry lines (7.1.5) through (7.1.8). Note that all symbols in equations (7.1.5) through (7.1.8) represent constants. To post these constants to the ledger, one just modifies the constant part of the formula already recorded there. For instance, after
posting (7.1.3) and (7.1.6), the debit balance in Depreciation Expense is

\[
\text{Balance} = \frac{a}{365 \cdot n} (d - d_0) - \frac{a}{365 \cdot n} (d_1 - d_0)
\]

(line) (7.1.3) (7.1.6)

\[
= \frac{a}{365 \cdot n} (d - d_1).
\]

After posting (7.1.8), the debit balance will be \( \frac{a}{365 \cdot n} (d - d_2) \). This pattern repeats itself year after year until the asset is sold or is fully written off. Sale of this asset will be considered shortly, in Figure 7.3.

*** Figure 7.2 goes approximately here ***

Figure 7.2 uses Lotus 123 formulae to provide a concrete illustration of the journal entries in Figure 7.1. Data for the example are given at the top of Figure 7.2. Journal entries (7.2.1) through (7.2.6) are a 123 implementation of journal entries (7.1.1) through (7.1.6) in Figure 7.1. In Lotus 123 the integer representation of any day (\( d \) in Figure 7.1) is found by evaluating the function \( \text{@int(@now)} \). \( @now \) examines the computer's system clock and returns a real number. \( @int \) converts that real number to an integer. On 23 March, 1990, both \( @int(@now) \) and \( @date(90,3,23) \) evaluate to the same number\(^2\). On subsequent days \( @int(@now) \) grows by 1 each day. Thus depreciation is incremented daily, in steps of $10. On 1 January, 1991, when line (7.2.6) is posted to the Depreciation Expense account, that account's balance will be

\(^2\) \( @date \) counts days from 1 January, 1900, so \( @date(90,3,23) \) evaluates to 32,955 days since 1 January, 1900.
Data: On 23 March 1990, a company acquires a machine for $36,500. The machine is to be depreciated for 10 years using straight-line depreciation with zero residual value. Financial year-ends are at 24:00 hours on 31 December 1990, 31 December 1991, etc..

Lotus 123 Implementation of lines (7.1.1) through (7.1.6) from Figure 7.1

23 March, 1990 (d_0):
Dr Asset (A) 36,500 (7.2.1)
Cr Bank 36,500 (7.2.2)
Dr Depreciation Expense (C) 10*(@int(@now)-@date(90,3,23)) (7.2.3)
Cr Accum. Depreciation (B) 10*(@int(@now)-@date(90,3,23)) (7.2.4)

00:01 hours, 1 January, 1991 (d_1):
Dr Profit and Loss 10*(@date(91,1,1)-@date(90,3,23)) (7.2.5)
Cr Depreciation Expense (C) 10*(@date(91,1,1)-@date(90,3,23)) (7.2.6)

Figure 7.2: A Concrete Example of Figure 7.1's Journal Entries

7-5(a)
Balance = 10\*(@int(@now)-@date(90,3,23)) - 10\*(@date(91,1,1)-@date(90,3,23))

(7.2.3)  (7.2.6)

= 10\*(@int(@now)-@date(91,1,1)).

Throughout 1 January, 1991, the above formula evaluates to zero.

The behaviour of the asset's account balances over the first few years of its life is shown in the graph in Figure 7.2. The horizontal axis measures time in Financial Years from 1 January, 1990. The balance in the Asset at Cost account (line A on the graph) does not change. Accumulated Depreciation (line B) rises steadily from the date of acquisition (i.e., from about 0.25 years into 1990). Depreciation Expense (line C) has a saw-tooth pattern. During each financial year it rises at the same rate as Accumulated Depreciation, but it drops to zero at the end of each financial year.

*** Figure 7.3 goes approximately here ***

Finally, to complete the example, suppose that sometime in the year after d_2, on day d_3, the asset is sold for $b. Figure 7.3 shows a set of five journal entries to record the sale of asset. Lines (7.3.1) and (7.3.2) are identical in structure to lines (7.1.5) and (7.1.6). They transfer the asset's current depreciation expense to Profit and Loss. Lines (7.3.3) through (7.3.8) transfer the balance formulae from the three accounts involved (i.e., the asset, accumulated depreciation, and depreciation expense for THIS asset) to the Sale of Asset account. After posting FA journal entries (7.3.3) through (7.3.10), the final balance of the Sale
**Data:** Data are the same as in Figure 7.1. The asset purchased on day $d_0$ has been depreciated for the years ending $d_1$ and $d_2$, and is sold on day $d_3$ for $b$.

**Five FA journal entries for day $d_3$:**

- **Dr Profit and Loss**
  - a
  - Cr Depreciation Expense
  - $a/(365 \times n) \times (d_3 \times d_2)$
  - (7.3.1)

- **Dr Sale of Asset**
  - a
  - Cr Asset
  - $a$  
  - (7.3.2)

- **Dr Accumulated Depreciation**
  - a
  - Cr Sale of Asset
  - $a/(365 \times n) \times (d_3 \times d_0)$
  - (7.3.3)

- **Dr Sale of Asset**
  - a
  - Cr Depreciation Expense
  - $a/(365 \times n) \times (d_3 \times d_0)$
  - (7.3.4)

- **Dr Cash**
  - b
  - Cr Sale of Asset
  - $b$  
  - (7.3.5)

---

**Figure 7.3:** HC/GAAP Formula Accounting for sale of fixed asset

7-6(a)
of Asset account (which could have been entered in place of the four journal
entry lines to Sale of Asset in Figure 7.3) is

\[
\text{Balance} = a - \frac{a}{(365*n)}(d-d_0) + \frac{a}{(365*n)}(d-d_3) - b
\]

(line) \hspace{1cm} (7.3.3) \hspace{1cm} (7.3.6) \hspace{1cm} (7.3.7) \hspace{1cm} (7.3.10)

\[
= a - \frac{a}{(365*n)}(d_3-d_0) - b.
\]

This is just the cost of the asset, \(a\), less the depreciation since date of purchase
at rate \(\frac{a}{(365*n)}\) per day, less disposal value, \(b\). Note that the time-dependent
variable, \(d\), in this balance has cancelled out; none of the remaining symbols is
time-dependent.

Generalising from this first example, any asset or liability where the account
balance depends predictably on the passage of time is a candidate for
simplification under FA. In such cases the computer simply interrogates its
internal clock at the start of report generation, then uses that time-value when
evaluating time-dependent balances for its reports.

7.1.2 Formulae Based on Current Market Values

*** Figure 7.4 goes approximately here, see next page ***

Accounting standards in many countries, e.g., SSAP 20 [ASC, 1983] and SFAS 52
[FASB, 1981], require foreign currency monetary assets and liabilities to be
valued at 'closing' exchange rates [IASC, 1983, IAS 21 paragraphs 32(a) and
Data: On day \( d_0 \), goods were purchased for US\$a. The foreign exchange rate (purchase price) for US dollars, \( x \), had value \( x_0 \) on day \( d_0 \), so that the purchase price in units of the home currency was \( a*x_0 \). On day \( d_1 \), sometime before year end, when the US\$a account was paid, the exchange rate had value \( x_1 \). Exchange rates are available to the FA ledger system through an on-line telephone link.

Formula Accounting Journal Entries

\[
\begin{align*}
\text{Day } d_0 & \\
\text{Dr Inventory} & a*x_0 & (7.4.1) \\
\text{Dr Foreign Exchange Gain/Loss} & a*(x-x_0) & (7.4.2) \\
\text{Cr Accounts Payable} & a*x & (7.4.3) \\
\text{Day } d_1 & \\
\text{Dr Accounts Payable} & a*x & (7.4.4) \\
\text{Cr Foreign Exchange Gain/Loss} & a*(x-x_1) & (7.4.5) \\
\text{Cr Bank} & a*x_1 & (7.4.6)
\end{align*}
\]

Figure 7.4: Accounting for a foreign currency liability
34(a)]. Figure 7.4 shows two transactions involving US dollars. The gains and losses recognized by line (7.4.2) are realized on day \(d_1\) when the account is paid and line (7.4.5) is posted to the ledger. On that day the debit balance in the Foreign Exchange Gain/Loss account is

\[
\text{Balance} = a(x - x_0) - a(x - x_1) \\
\text{(line) (7.4.2) (7.4.5)} \\
= a(x_1 - x_0).
\]

If the movement in the exchange rate has been unfavourable, this debit balance will be positive, i.e., a loss will have been realized.

It is important to note that the two journal entries in Figure 7.4, i.e., lines (7.4.1) through (7.4.3), and (7.4.4) through (7.4.6), stay balanced over time. Since the exchange rate, \(x\), can fluctuate in quite unpredictable ways, the only way the journal entries can stay balanced over time is if the sum of the debit coefficients of variable \(x\) in each journal entry exactly matches the sum of credits. For instance in lines (7.4.1) through (7.4.3), the debit and credit coefficients of \(x\) are both \(a\). So the journal entry will stay in balance no matter how much \(x\) varies. Where there is more than one variable in a journal entry, provided debit and credit coefficients of each variable in each journal entry are equal, the ledger will stay in balance over time.

Generalising from this second example, accounting for any asset or liability where the account balance depends on current market prices is also a candidate for simplification under FA. This includes shares traded on stock markets, commodities like gold, copper, or wheat, and liabilities like bonds and
debentures (where market interest rates change). On a practical note, the example above assumed on-line access to market prices, but as few unrealized revaluations are reported in HC/GAAP accounting systems, dial-up access to market prices, or even periodic entry of newspaper data, might be quite satisfactory for many firms.

7.1.3 Formulae Based on Other Account Balances

*** Figure 7.5 goes approximately here, see next page ***

One of the most powerful features of spreadsheet programs is the way that a formula in one cell can refer to the contents of other cells. Similarly in a FA general ledger system it seems desirable that a formula in one account should be able to refer to balances of other ledger accounts. Figure 7.5 illustrates one such application. The mechanism used here for referring to the balance of an account is to place the account name in single quotes. After posting lines (7.5.1) and (7.5.2) the balance of the Allowance for Doubtful Debts account will always be 5% of the Accounts Receivable account balance. At year end the balance in Doubtful Debts Expense would be transferred to Profit and Loss by lines (7.5.3) and (7.5.4) leaving the balance of the expense account at the start of the next period as:
Data: On day $d_0$, the balances of the Allowance for Doubtful Debts and Doubtful Debts Expense accounts, are $a_0$ (credit balance) and $e_0$ (debit balance), respectively. From that time onwards, the Allowance for Doubtful Debts is to be set to 5% of the current Accounts Receivable balance.

**Formula Accounting Journal Entries**

<table>
<thead>
<tr>
<th>Day $d_0$</th>
<th>Dr Doubtful Debts Expense</th>
<th>A/C Receivable$^0.05-a_0$ (7.5.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cr Allowance for Doubtful Debts</td>
<td>A/C Receivable$^0.05-a_0$ (7.5.2)</td>
<td></td>
</tr>
</tbody>
</table>

At year end

<table>
<thead>
<tr>
<th>Dr Profit and Loss</th>
<th>A/C Receivable$^0.05-a_0+e_0$ (7.5.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cr Doubtful Debts Expense</td>
<td>A/C Receivable$^y/e^0.05-a_0+e_0$ (7.5.4)</td>
</tr>
</tbody>
</table>

Note: 'A/C Receivable' means 'the current balance of the Accounts Receivable account', and 'A/C Receivable$^y/e$' means the balance of the account at financial year end after $d_0$. (The single-quotes-around-the-account-name notation is borrowed from the financial modelling package FCS-EPS.)

Figure 7.5: Allowance for Doubtful Debts as 5% of Accounts Receivable
Balance = $e_0 + (\text{A/C Receivable} \times 0.05 - a_0) - (\text{A/C Receivable}\text{'}_{y/e} \times 0.05 - a_0 + e_0)$

\[ \text{(line) } \quad \text{(7.5.1)} \quad \text{(7.5.4)} \]

\[ = (\text{A/C Receivable} - \text{A/C Receivable}\text{'}_{y/e}) \times 0.05 \]

i.e., the expense next year will be 5% of the change in the Accounts Receivable account balance after year end.

It is tempting to try to use formulae based on other account balances to automate cost allocations and end-of-period account closure. However, formulae do not transfer account balances from one account to another. For instance, one could try to allocate plant-wide Building Services overhead to Departments A and B as follows:

\begin{align*}
\text{Dr} & \quad \text{Overhead, Department A} & \quad \text{Building Services} \times 0.6 \\
\text{Dr} & \quad \text{Overhead, Department B} & \quad \text{Building Services} \times 0.4 \\
\text{Cr} & \quad \text{Contra - Building Services} & \quad \text{Building Services}
\end{align*}

Here it is true that as costs are charged to the Building Services account during the next period, the balances in the two Overhead accounts will correctly reflect their share of the costs. It is also true that the net balance of the Contra account and the Building Services account will always equal zero\(^3\). However, the journal entry above does not transfer the balance of Building

---

\(^3\) A Contra account is necessary. Crediting Building Services with its own balance would result in what spreadsheet systems usually call a circular reference. If the circular reference were defined to have meaning (so that the balance in Building Services reduced to zero) the two debit entries would also be zero, which is not what is intended.
Services to the Departmental Overhead accounts; the balance in Building Services will go on rising for as many years as the account exists in the ledger. The same comment applies to any other attempts to 'transfer' or 'close' account balances using formulae based on this account-balance notation. Unless the ledger system has some housekeeping procedure that can convert account-balance formula transfers into real ones, e.g., at the end of each period, it seems better to transfer and/or close off all revenue and expense accounts with explicit FA journal entries each period. These could, of course, be generated automatically.

7.2 Examples of Inflation/Current value Accounting

Figures 7.1 through 7.5 in Section 7.1 show how FA could be used for automatic revision of asset and liability values in an HC/GAAP system. However, Formula Accounting was developed specifically to meet Design Principle 1 from Chapter 6. This section shows how it works. Section 7.2.1 below considers the CC/CPP equivalent of the fixed asset accounting in Section 7.1. Section 7.2.2 presents FA journal entries for CC/CPP trading accounts. Section 7.2.3 shows how to account for a decision to change the price index series on which the asset is valued.

7.2.1 CC/CPP Fixed Asset Accounting

The examples in Figures 7.6 through 7.8 are based on the same data as the example in Figures 7.1 and 7.3, but this time CC/CPP accounting is used. In this example, the value of the fixed asset is indexed by a specific price index
series, s, and gains (losses) relative to a general inflation index, g, are credited to a Holding Gain/Loss account. Since the asset is acquired for cash, there are also holding gains and losses on the change in the cash balance (often called Purchasing Power gains and losses) to be accounted for. To place these in a plausible context it is assumed initially, with no loss of generality, that the business started on day $d_{-1}$, when the owner contributed $c$ to the business.

*** Figure 7.6 goes approximately here, see next page ***

In the Formula Accounting journal entries in Figure 7.6, line (7.6.1) records the cash contributed by the owner. Line (7.6.3) records owners' equity, which is indexed by the general level of prices. Line (7.6.2) is the balancing debit; it records the loss of purchasing power through holding cash. (In this example, all revaluations, including gains and losses on monetary items, are posted to the one account, a control account called Holding Gain/Loss.) Notice that the only variable in lines (7.6.1) through (7.6.3) is $g$ in lines (7.6.2) and (7.6.3). Values for price indices, g and s, have to be supplied to the general ledger system when they become available. It is assumed that suitable government or special index series (e.g., manufacturers' price lists for inventory) are available, and that the FA ledger system extrapolates from the prior time series whenever an index value is needed. A cruder alternative to extrapolation would be to use the latest index value until the next one became available.

7.2.1(a) Purchase and Depreciation of Fixed Asset

Formula Accounting journal entries (7.6.4) through (7.6.9) in Figure 7.6 are the
Data: On day $d_{-1}$ a firm is established with $c$ cash. Soon after, on day $d_0$, a fixed asset is acquired for $a$. The asset's expected life is $n$ years and its residual value is expected to be zero. Straight-line depreciation is considered appropriate. The end of the first financial year after asset acquisition is day $d_1, (d_1-d_0)\leq 365$. The end of the next financial year is $d_2$. The following are the relevant price index series (all subscripted symbols are constants):

<table>
<thead>
<tr>
<th>day</th>
<th>$d_{-1}$</th>
<th>$d_0$</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>specific prices</td>
<td>$s_{-1}$</td>
<td>$s_0$</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s$</td>
</tr>
<tr>
<td>general prices (CPI)</td>
<td>$g_{-1}$</td>
<td>$g_0$</td>
<td>$g_1$</td>
<td>$g_2$</td>
<td>$s$</td>
</tr>
</tbody>
</table>

Formula Accounting Journal Entries

**Day $d_{-1}$:**
Dr Bank  
Dr Holding Gain/Loss  
Cr Capital  

\[\begin{align*}
\text{Dr Bank} & \quad c \\
\text{Dr Holding Gain/Loss} & \quad c(g/g_{-1} - 1) \\
\text{Cr Capital} & \quad c(g/g_{-1})
\end{align*}\]  
(7.6.1)
(7.6.2)
(7.6.3)

**Day $d_0$:**
Dr Asset  
Cr Holding Gain/Loss  
Cr Bank  
Cr Holding Gain/Loss  

\[\begin{align*}
\text{Dr Asset} & \quad a(s/s_0) \\
\text{Cr Holding Gain/Loss} & \quad a(s/s_0 - g/g_0) \\
\text{Cr Bank} & \quad a \\
\text{Cr Holding Gain/Loss} & \quad a(g/g_0 - 1)
\end{align*}\]  
(7.6.4)
(7.6.5)
(7.6.6)
(7.6.7)

Dr Depreciation Expense  
Cr Accumulated Depreciation  

\[\begin{align*}
\text{Dr Depreciation Expense} & \quad a(s/s_0)/(365*n) \times (d-d_0) \\
\text{Cr Accumulated Depreciation} & \quad a(s/s_0)/(365*n) \times (d-d_0)
\end{align*}\]  
(7.6.8)
(7.6.9)

**Day $d_1$:**
Dr Profit and Loss  
Dr Holding Gain/Loss  
Cr Depreciation Expense  

\[\begin{align*}
\text{Dr Profit and Loss} & \quad B_1*(g/g_{-1}) \\
\text{Dr Holding Gain/Loss} & \quad B_1*(s/s_1 - g/g_1) \\
\text{Cr Depreciation Expense} & \quad B_1*(s/s_1)
\end{align*}\]  
(7.6.10)
(7.6.11)
(7.6.12)

(Where $B_1$ is $a(s/s_0)/(365*n)*(d_1-d_0)$, a constant, being the current balance in the Depreciation Expense account on day $d_1$.)

**Day $d_2$:**
Dr Profit and Loss  
Dr Holding Gain/Loss  
Cr Depreciation Expense  

\[\begin{align*}
\text{Dr Profit and Loss} & \quad B_2*(g/g_2) \\
\text{Dr Holding Gain/Loss} & \quad B_2*(s/s_2 - g/g_2) \\
\text{Cr Depreciation Expense} & \quad B_2*(s/s_2)
\end{align*}\]  
(7.6.13)
(7.6.14)
(7.6.15)

(Where $B_2$ is $a(s_2/s_0)/(365*n)*(d_2-d_1)$, a constant, being the current balance in the Depreciation Expense account on day $d_2$.)

Figure 7.6: CC/CPP Formula Accounting for fixed assets

7-12(a)
CC/CPP equivalents of lines (7.1.1) through (7.1.4) in the HC/GAAP system (Section 7.1.1). It has been assumed in framing the FA journal entries that prices are rising, with specific price rises, s, exceeding general price rises, g. As with the exchange rates example in Figure 7.4, if the value of a formula is negative then debit becomes credit, and vice versa⁴. Lines (7.6.4) and (7.6.5) record the fixed asset at its current value, and the real holding gain, i.e., the gain above the general level of prices, respectively. Note that line (7.6.5) is a linear function of both s and g. Lines (7.6.6) and (7.6.7) have the opposite effect to lines (7.6.1) and (7.6.2). If a is greater than c, the firm will start making purchasing power gains on net monetary liabilities. Lines (7.6.8) and (7.6.9) are the logical equivalents of (7.1.3) and (7.1.4), but now they are both functions of s and d, not d alone.

Already, with this single asset example, the most complex account balance formula is that of the Holding Gain/Loss account. In CC/CPP accounting, this is the real holding gain/loss, after allowing for changes in the general level of prices. The formula for the balance, positive for debit, negative for credit, is presently as follows:

---

⁴ It is awkward to have to think of negative debits and negative credits. As a notation, Debits and Credits seem to assume that we are recording things that happened in the past, so that the debits and credits always apply to positive numbers. However, FA journal entries have to allow for future changes in asset values, where the possibility of negative entries is quite real. In many cases it may be simpler to frame journal entries in terms of plus and minus debit formulae only, and to check that journal entries add to zero rather than to check that they balance.
\[
\text{Balance} = \frac{c(g/g_0 - 1)}{(1)} - \frac{a(s/s_0 - g/g_0)}{(7.6.5)} - \frac{a(g/g_0 - 1)}{(7.6.7)}
\]

\[
= \frac{c(g/g_{-1} - 1)}{1} - \frac{a(s/s_0 - 1)}{(7.6.2)}
\]

In this example, the balance simplifies to a function of only two variables, \(s\) and \(g\). Generalising to the multi-asset firm, the Holding Gain/Loss account would probably be the most complex to evaluate, because it would depend on all the different specific price index series being used in the accounting system.

Lines (7.6.10) through (7.6.15) are the CC/CPP equivalents of lines (7.1.5) through (7.1.8). Line (7.6.10) transfers day \(d_1\)'s depreciation expense figure (a constant) to Profit and Loss, then indexes it by the general level of prices. Line (7.6.12) is analogous to line (7.1.6). After posting it the formula for the Depreciation Expense balance will be:

\[
\text{Balance} = \frac{a(s/s_0)}{(365*n)}*(d-d_0) - \frac{a(s_1/s_0)}{(365*n)}*(d_1-d_0)*(s/s_1)
\]

\[
= \frac{a(s/s_0)}{(365*n)}*(d-d_1).
\]

Line (7.6.11) balances the journal entry. With the asset now partly written down, line (7.6.11) ensures holding gains recorded from day \(d_1\) onwards will be smaller. Lines (7.6.13) to (7.6.15) follow exactly the same pattern at the end of the second year, on day \(d_2\).
The final task before leaving this section is to check that all five journal entries in Figure 7.6 stay balanced over time. In lines (7.6.1) through (7.6.3), the only variable is \( g \), and the debit and credit coefficients of \( g \) are both \( c/g_{-1} \). So the journal entry will stay in balance as \( g \) varies. In lines (7.6.4) through (7.6.7) the debit and credit coefficients of \( s \) are both \( a/s_0 \), and of \( g \) are both zero. So the journal entry will stay in balance. In lines (7.6.8) and (7.6.9) the coefficients of \( s \) and \( s^d \) are obviously equal (the debit and credit formulae are identical), so the journal entry will stay in balance. In lines (7.6.10) through (7.6.12), the debit and credit coefficients of \( s \) are \( B_1/s_1 \), and of \( g \) are zero. Likewise for lines (7.6.13) through (7.6.15). Thus all five journal entries in Figure 7.6 will stay in balance over time.

7.2.1(b) Alternative treatment of depreciation in CC/CPP Formula Accounting

Purist current-value accountants, such as Chambers [1966], would probably say that it is foolish to try to maintain separate accounts for the current cost of an equivalent new asset, and its accumulated depreciation. In many ways they would be right. In a world of ever-changing technology, replacement costs, and even net realizable values, of new assets will be very difficult to estimate. For this reason purist current-value accountants often recommend attempting to find the replacement cost or net realizable value of the particular used asset owned by the firm, and calling the change in value depreciation. If that procedure were followed, the 'written down' value of the asset would be determined by a new index, say \( s' \), which would normally get smaller as the asset was 'used up'. There would be no concept of a Holding gain/loss on the fixed asset. Nor would there be any concept of Accumulated Depreciation.
entries (7.6.4) through (7.6.9) would be replaced by the entries shown in Figure 7.7.

*** Figure 7.7 goes approximately here, see next page ***

There are two comments about the journal entry in Figure 7.7. First, FA imposes no restrictions on the expressive power of the accountant. FA is just as comfortable with lines (7.7.1) through (7.7.4) as it was with lines (7.6.4) through (7.6.9). Second, as discussed in more detail in Section 7.2.2(b), it seems likely that the income measure, Current Cost Income from Continuing Operations (CCIFCO), resulting from (7.7.2) would be more volatile than if line (7.6.8) were used. All other things being equal, line (7.6.8) smooths changes in value over longer periods. If CCIFCO is being used as an indicator of future average earning power, then line (7.6.8) is likely to result in more useful information. Clearly, either approach can be used in FA. The sale-of-asset example below retains the Accumulated Depreciation approach.

7.2.1(c) Sale of fixed asset in CC/CPP Formula Accounting

The CC/CPP equivalent of Figure 7.3 is shown in Figure 7.8. In addition to the data in Figure 7.3, on \( d_3 \) the specific and general price indices are \( s_3 \) and \( g_3 \). The CC/CPP journal entry for the sale of asset is shown in Figure 7.8(a), with \( B_4 \) in line (7.8.10) defined as the credit balance from the T-account in Figure 7.8(b). As in conventional accounting, details for the journal entries come from a fixed asset register of some kind. Using CC/CPP Formula Accounting for this sale of asset, information required from the fixed asset register will be (i) the
Data: Data are the same as in Figure 7.6, but the 'written down' value of the asset is determined by a new index, $s'$, which normally gets smaller as the asset is 'used up'.

FAJE for day $d_0$:

<table>
<thead>
<tr>
<th>Account</th>
<th>Entry</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr Asset</td>
<td>$a(s'/s_0')$</td>
<td>(7.7.1)</td>
</tr>
<tr>
<td>Dr Depreciation Expense</td>
<td>$a(g/g_0) - a(s'/s_0')$</td>
<td>(7.7.2)</td>
</tr>
<tr>
<td>Cr Bank</td>
<td>$a$</td>
<td>(7.7.3)</td>
</tr>
<tr>
<td>Cr Holding gain/loss</td>
<td>$a(g/g_0 - 1)$</td>
<td>(7.7.4)</td>
</tr>
</tbody>
</table>

Figure 7.7: Depreciation based on the declining value of the used asset
formula for the current cost, (ii) the formula for accumulated depreciation, (iii) the formula for depreciation expense for the current period, and (iv) the formula for net holding gain/loss.

*** Figure 7.8 goes approximately here, see next page ***

Lines (7.8.1) through (7.8.3) are identical in format to lines (7.6.10) through (7.6.12). They transfer the asset's current depreciation expense to Profit and Loss. On day \( d_3 \), after line (7.8.3) has been posted, the balance in the asset's Depreciation Expense account would display on a computer screen as zero, but the formula for the balance would be \( a(s/s_0)/(365*n)*(d-d_3) \).

Lines (7.8.4) through (7.8.11) transfer the balance formulae from the four accounts involved (i.e., the asset, accumulated depreciation, depreciation expense, and holding gain/loss, all for THIS asset only) to a Sale of Asset account. Details of the Holding gain/loss account are shown in Figure 7.8(b). This is a case where the symbolic representation of the account balance is more complex than the evaluated expression; when all the symbols for constants are replaced by real numbers, the balance is just a linear function of \( s \) and \( g \). Lines (7.8.12) through (7.8.14) show the owner receiving cash, \( b \), which in times of inflation, will erode in purchasing power. The balance in the Sale of Asset account is non-operating income; it is a linear function of the general index, \( g \).
Data: Data are the same as in Figure 7.6. The asset purchased on $d_0$ has been depreciated for the years ending $d_1$ and $d_2$, and is sold on day $d_3$ for $b$. Specific and general indices on day $d_3$ are $s_3$ and $g_3$ respectively.

Six FAJEs for day $d_3$:

<table>
<thead>
<tr>
<th>Debit</th>
<th>Credit</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr Profit and Loss</td>
<td>B_3 * (g/g_3)</td>
<td>(7.8.1)</td>
</tr>
<tr>
<td>Dr Holding gain/loss</td>
<td>B_3 * (s/s_3 - g/g_3)</td>
<td>(7.8.2)</td>
</tr>
<tr>
<td>Cr Depreciation Expense</td>
<td>B_3 * (s/s_3)</td>
<td>(7.8.3)</td>
</tr>
</tbody>
</table>

(Where $B_3 = a(s_3/s_0)/(365*n) * (d_3-d_2)$, a constant.)

<table>
<thead>
<tr>
<th>Debit</th>
<th>Credit</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr Sale of Asset</td>
<td>a(s/s_0)</td>
<td>(7.8.4)</td>
</tr>
<tr>
<td>Cr Asset</td>
<td>a(s/s_0)</td>
<td>(7.8.5)</td>
</tr>
<tr>
<td>Dr Accumulated Depreciation</td>
<td>a(s/s_0)/(365*n) * (d-d_0)</td>
<td>(7.8.6)</td>
</tr>
<tr>
<td>Cr Sale of Asset</td>
<td>a(s/s_0)/(365*n) * (d-d_0)</td>
<td>(7.8.7)</td>
</tr>
<tr>
<td>Dr Sale of Asset</td>
<td>a(s/s_0)/(365*n) * (d-d_3)</td>
<td>(7.8.8)</td>
</tr>
<tr>
<td>Cr Depreciation Expense</td>
<td>a(s/s_0)/(365*n) * (d-d_3)</td>
<td>(7.8.9)</td>
</tr>
<tr>
<td>Dr Holding gain/loss</td>
<td>B_4 (see below)</td>
<td>(7.8.10)</td>
</tr>
<tr>
<td>Cr Sale of Asset</td>
<td>B_4 (see below)</td>
<td>(7.8.11)</td>
</tr>
<tr>
<td>Dr Cash</td>
<td>b</td>
<td>(7.8.12)</td>
</tr>
<tr>
<td>Dr Holding gain/loss</td>
<td>b(g/g_3 - 1)</td>
<td>(7.8.13)</td>
</tr>
<tr>
<td>Cr Sale of Asset</td>
<td>b(g/g_3)</td>
<td>(7.8.14)</td>
</tr>
</tbody>
</table>

After posting the above FAJEs, the final (credit) balance of the Sale of Asset account is a linear function of $g$: $-a(g/g_0) + b(g/g_3) + a/(365*n) * [(s_1/s_0)(d_1-d_0)(g/g_1) + (s_2/s_0)(d_2-d_1)(g/g_2) + (s_3/s_0)(d_3-d_2)(g/g_3)]$

Figure 7.8(a): CC/CPP FAJEs for sale of fixed asset

Balance, $B_4$, in lines (7.8.10) and (7.8.11) is determined as the credit balance in the following T-account. The journal entry numbers beside each entry are based on FAJEs in Figure 7.6 and line (7.8.2). The resultant balance formula is a linear function of $s$ and $g$.

<table>
<thead>
<tr>
<th>Holding gain/loss (FOR THIS ASSET ONLY)</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>(7.6.11)</td>
<td>a(s_1/s_0)/(365*n) * (d_1-d_0) * (s/s_1 - g/g_1)</td>
</tr>
<tr>
<td>(7.6.14)</td>
<td>a(s_2/s_0)/(365*n) * (d_2-d_1) * (s/s_2 - g/g_2)</td>
</tr>
<tr>
<td>(7.8.2)</td>
<td>a(s_3/s_0)/(365*n) * (d_3-d_2) * (s/s_3 - g/g_3)</td>
</tr>
<tr>
<td>(7.6.5)</td>
<td>a(s/s_0 - g/g_0)</td>
</tr>
</tbody>
</table>

Figure 7.8(b): Calculating the balance, $B_4$, in the CC/CPP Holding gain/loss account for this fixed asset

Figure 7.8: CC/CPP Formula Accounting for sale of fixed asset

7-17(a)
7.2.2 Trading accounts in CC/CPP Formula Accounting

7.2.2(a) Choices concerning income measurement

The next example involves purchase and sale of inventory in CC/CPP FA. Before working through the example, however, there are three matters concerning income measurement to be discussed. First, in an articulated double-entry ledger system one can always calculate comprehensive income (SFAC 3 [FASB, 1980]) as either (i) the change in 'wealth', i.e., net assets, between two periods of time, or (ii) by matching expenses against revenues, then adding holding gains and losses\(^5\). The two methods should always result in the same comprehensive income measure with the same measurement error, and often it is less effort to compare net assets at two periods of time. However, the change-in-wealth approach does not yield a measure of current cost income from continuing operations, CCIFCO. In the belief that CCIFCO is a useful indicator of future earning capacity, possibly subject to smaller measurement error than comprehensive income\(^6\), the recording system below follows convention and calculates CCIFCO and holding gains separately. They are then added, to yield comprehensive income.

\(^5\) Of course, to an economist, this "comprehensive" income is not comprehensive at all; it does not measure the change in net present values of a firm's future cash flows. However, because of uncertainty about estimates of future cash flows, it is generally agreed that a reliable measure of economic income is unattainable.

\(^6\) In the US the SEC toyed with a change-in-wealth approach to income measurement in its Reserve Recognition Accounting for the Oil and Gas industry [Wolk, Francis and Tearney, 1984, pp.463-466]. However, the measurement error associated with valuing proved oil reserves at the end of each year was so great that the experiment had to be abandoned.
Second, it is customary for standards on CCA to require that sales and expenses should be calculated in middle-of-period money units. Justification given for this approach is that income is a measure of changes in wealth during a period and it is therefore appropriate to report income in middle-of-period money units. One could tolerate such suggestions if the resultant income measure were then restated to end-of-period money units before being added to Retained Earnings. But many CCA systems ignore the changing purchasing power of the monetary unit. They add the middle-of-period money units to start-of-period units, and compound the error by reporting the result on the end-of-period balance sheet. This is quite unsatisfactory. In the CC/CPP system developed below, therefore, all reports will be expressed in end-of-period money units.

Third, reporting in end-of-period units means that sales revenue and all expenses have to be restated. There are two possible ways of restating sales and expenses. One approach would be to record sales and expenses of all different types of resources in physical units, then to apply end-of-period prices to those physical units. However, this would lead to problems if selling margins changed markedly during the period. Since it is the margins that are of primary interest in calculating CCIFCO, this 'valuation at end of period' approach must be rejected. Better matching is effected if one records the money value of sales and current costs of expenses on the date they occur, then restates those figures for the general level of price change from that date to report date\(^7\). Conceptually this is no different to (i) calculating the profit on every sale or expense as the transaction occurs, then (ii) restating that profit, which is now

---

\(^7\) Report date is the day the account balances are displayed on a screen, or printed on a report.
owners' equity, for general price changes. Thus all CC/CPP revenue and expense transactions in the example below are restated from event date to report date by general price indices.

7.2.2(b) Purchase and sale of inventory in CC/CPP Formula Accounting

Figure 7.9 is a simple example of purchase and sale of inventory journal entries in CC/CPP FA\(^8\). It also illustrates the three points above. Under the change-in-wealth approach (Solution 1, near the top of Figure 7.9), income is simply the wealth at end, \(b\), less the wealth at the start restated for the general level of prices, \(a(g/g_0)\). As shown in the Income statement in Solution 2 (in the lower half of Figure 7.9) the matching approach begins by calculating current cost income from continuing operations, CCIFCO. When real holding gains and losses are added to CCIFCO, the resulting CC/CPP income measure (the bottom line in Figure 7.9) is identical to income calculated using the difference-in-wealth approach, Solution 1.

*** Figure 7.9 goes approximately here, see next page ***

Journal entry lines (7.9.1) through (7.9.3) are conceptually equivalent to lines (7.6.1) through (7.6.3) in Figure 7.6. If the firm had continued to hold cash it would have made purchasing power losses at the rate shown in line (7.9.2). Similarly, the purchase journal entry, lines (7.9.4) through (7.9.7), is conceptually equivalent to lines (7.6.4) through (7.6.7) of Figure 7.6. However,

\(^8\) Symbols \$a and \$b (and \$c in s.7.2(c)) are unrelated to the previous examples.
Data: On day $d_0$, a firm is established with $a$ cash. The same day inventory is purchased for cash amount $a$. Later, on day $d_1$, the inventory is sold to customer $x$, on credit, for $b$. (All subscripted symbols are constant. Symbols $a$, $b$, and $c$ are unrelated to the previous examples. Symbols $g$ and $s$ are as defined in Figure 7.6.)

Required: Calculate CC/CPP income for the period from $d_0$ to $d$, $d>d_1$.

Solution 1:
Using the change-in-wealth approach, CC/CPP income on day $d = b - a(g/g_0)$

Solution 2:
One way of matching revenue and expenses is as follows:

\begin{align*}
\text{Starting capital:} \\
\text{Dr Bank} & : a & (7.9.1) \\
\text{Dr Holding gain/loss} & : a(g/g_0 - 1) & (7.9.2) \\
\text{Cr Owners' Equity} & : a(g/g_0) & (7.9.3) \\
\text{Purchase of inventory:} \\
\text{Dr Inventory} & : a(s/s_0) & (7.9.4) \\
\text{Cr Holding gain/loss} & : a(s_0/g,g_0) & (7.9.5) \\
\text{Cr Bank} & : a & (7.9.6) \\
\text{Cr Holding gain/loss} & : a(g/g_0 - 1) & (7.9.7) \\
\text{Sale of inventory:} \\
\text{Dr Accounts receivable, } x & : b & (7.9.8) \\
\text{Dr Holding gain/loss} & : b(g/g_1 - 1) & (7.9.9) \\
\text{Cr Sales} & : b(g/g_1) & (7.9.10) \\
\text{Dr Cost of sales} & : a(s_1/s_0)(g/g_1) & (7.9.11) \\
\text{Dr Holding gain/loss} & : a(s_0/s_0 - a(s_1/s_0)(g/g_1)) & (7.9.12) \\
\text{Cr Inventory} & : a(s_0/s_0) & (7.9.13) \\
\end{align*}

Income statement, period $d_0$ to $d$

\begin{align*}
\text{Sales} & : b(g/g_1) & (7.9.10) \\
\text{less Cost of sales} & : a(s_1/s_0)(g/g_1) & (7.9.11) \\
\text{CCIFCO} & : (b-a(s_1/s_0)(g/g_1)) & (7.9.12) \\
\text{plus Holding gain on inventory} & : a(s_0/g_0 - g/g_0) & (7.9.5) \\
\text{Holding gain on mon. asset} & : -[a(s_0/g_0 - a(s_1/s_0)(g/g_1))] & (7.9.12) \\
\text{Comprehensive CC/CPP income} & : b - a(g/g_0) & (7.9.9) \\
\end{align*}

Figure 7.9: Calculating CC/CPP Income from a purchase and sale

7-20(a)
because inventory is not normally depreciated, there is no equivalent to lines (7.6.8) and (7.6.9) of Figure 7.6.

In the sale-of-inventory journal entry, lines (7.9.8) through (7.9.13), the new monetary asset and corresponding purchasing power loss are shown in line (7.9.8) and (7.9.9). The credit to Sales and debit to Cost of Sales are indexed by the general level of prices in lines (7.9.10) and (7.9.11). Finally, inventory is credited in line (7.9.13), with line (7.9.12) balancing inventory and cost of sales. After subtracting the formula in line (7.9.12) from that in line (7.9.5) the net holding gain due to holding inventory is \( a(\frac{s_1}{s_0} - \frac{g_1}{g_0}) (\frac{g}{g_1}) \), i.e., line (7.9.12) locks-in the real holding gain of \( a(\frac{s_1}{s_0} - \frac{g_1}{g_0}) \) set in motion by (7.9.5).

7.2.2(c) Aggregating FA journal entries

The purpose of this section is to show that as journal entries are posted to a FA general ledger, each new entry may be added to the formula for the existing balance in a quite predictable and manageable way. Note that the formulae in lines (7.9.1) through (7.9.13) contain only two variables, the price indices \( s \) and \( g \). As FAJEs like those shown in Figure 7.9 are posted to the ledger day after day, the values of the coefficients of \( s \) and \( g \) will change, and the values of the constant (subscripted) symbols will be different each day, but the number of variables will not. This means that a FA ledger system can be used for summarizing large numbers of transactions, just as ledger systems are supposed to do. The example works with CC/CPP journal entries, but the technique is quite general.
There are so many transactions (377 to be precise) in the following example that it is necessary to work with absolute numbers, not symbols, for the data. Assume that a firm buys 40 units of inventory at $500 each on 1 January. It sells one unit each day for $1000 each, and buys 30 new units on the first day of each calendar month. Thus, over the year the firm buys 370 units of inventory and sells 365. The general price index rises linearly from 100, on 1 January, to 112, on 31 December, i.e., slowly declining general inflation. The specific price index for the inventory also rises linearly from 100, on 1 January, to 124, on 31 December. To keep the index numbers simple, the latest monthly index value will be used for indexing purposes, i.e., the general index used in February will be 101 for all transactions in February. The objective is to find the formulae for all account balances on 31 December.

*** Figure 7.10 goes approximately here, see next page ***

The upper half of Figure 7.10 relates to January, the lower half to February. In both months, the one purchase FAJE, and one representative sale FAJE is shown. Each is based on the purchase and sale templates in Figure 7.9. All other FAJEs for the month are identical to the first. Note that for January the coefficient of $s$ in (7.10.9) is 40 times smaller than that in line (7.10.1), because 40 units of inventory were purchased, but only one unit was sold, on the first day. The trial balance for 31 January is the result of posting the one purchase journal entry, and 31 repetitions of the sale entry to the FA general ledger. Note that the formulae for the account balances are very simple functions of $s$ and $g$, and that the totals balance.
Purchase of inventory (1 January):

Dr Inventory 
Cr Holding gain/loss 
Cr Bank 
Cr Holding gain/loss 

Sale of inventory (1 January):

Dr Accounts receivable 
Dr Holding gain/loss 
Cr Sales 

Dr Cost of sales 
Cr Inventory 
Cr Holding gain/loss 

Trial Balance, 31 January

<table>
<thead>
<tr>
<th>Dr</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank</td>
<td>20000</td>
</tr>
<tr>
<td>Accounts receivable</td>
<td>31000</td>
</tr>
<tr>
<td>Inventory</td>
<td>45s</td>
</tr>
<tr>
<td>Sales</td>
<td>310g</td>
</tr>
<tr>
<td>Cost of sales</td>
<td>155g</td>
</tr>
<tr>
<td>Holding gain/loss</td>
<td></td>
</tr>
</tbody>
</table>

Trial Balance, 28 February

<table>
<thead>
<tr>
<th>Dr</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank</td>
<td>35300</td>
</tr>
<tr>
<td>Accounts receivable</td>
<td>59000</td>
</tr>
<tr>
<td>Inventory</td>
<td>55s</td>
</tr>
<tr>
<td>Sales</td>
<td>587.228g</td>
</tr>
<tr>
<td>Cost of sales</td>
<td>296.386g</td>
</tr>
<tr>
<td>Holding gain/loss</td>
<td></td>
</tr>
</tbody>
</table>

On the first day of February the firm buys 30 more units of inventory at a price of, say, $500*(102/100) = $510 each.

Purchase of inventory (1 February):

Dr Inventory 
Cr Holding gain/loss 
Cr Bank 
Cr Holding gain/loss 

Sale of inventory (1 February):

Dr Accounts receivable 
Dr Holding gain/loss 
Cr Sales 

Dr Cost of sales 
Cr Inventory 
Cr Holding gain/loss 

Trial Balance, 28 February

<table>
<thead>
<tr>
<th>Dr</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank</td>
<td>35300</td>
</tr>
<tr>
<td>Accounts receivable</td>
<td>59000</td>
</tr>
<tr>
<td>Inventory</td>
<td>55s</td>
</tr>
<tr>
<td>Sales</td>
<td>587.228g</td>
</tr>
<tr>
<td>Cost of sales</td>
<td>296.386g</td>
</tr>
<tr>
<td>Holding gain/loss</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.10 Aggregating FAJE's for January and February

7-22(a)
In February, sales continue at $1000 per day, but the general level of prices has risen, so the real value of sales is lower. This is reflected in line (7.10.17). In addition, the replacement cost of goods sold has been rising faster than the general level of prices, line (7.10.18), so real margins are even lower. The trial balance for 28 February shows the result of adding the February purchase, and 28 repetitions of the February sale FAJEts to the January account balances. The formulae are still no more complex than in January, and again, the accounts balance.

Aside: Note that on 1 February, before that day's sale, there are 9 units of inventory that cost $500 each a month ago, and 30 units that just cost $510. The balance in the Inventory account for these 39 units will be the opening balance (45s, from the Trial Balance) plus the new purchase (150s, line (7.10.11)), i.e., 195s. The question arises: How would the FAJEts be affected if the purchase price and the specific price index had not moved synchronously?

To answer this question, assume the new 30 units cost $505 each, not $510. By line (7.10.11) the debit to Inventory would be 30*\(\frac{505}{102}\) = 148.529s, so the new balance in the account would be 193.529s, not 195s. The unit current cost would then be 193.529s/39 = 4.96229s, not 5s. Evidently the formulae in lines (7.10.19) and (7.10.20), i.e., the credit to Inventory and adjustment to Holding gain, should always be expressed as the proportion of units about to be removed, multiplied by the current balance in inventory.

For March through December, journal entries continue like those in January and February. The numbers underlined in the February FAJEts increase each month as specific and general prices rise. The final trial balance, and Accounting
Reports for 31 December are shown in Figure 7.11. Note again that as more transactions are aggregated the coefficients of the various indices have grown, but there is no increase in the complexity of the formulae in the trial balance. Incrementing the coefficients as journal entries are posted, and evaluating the formulae for Income and Balance Sheet reporting is a quite manageable task for present-day computers.

*** Figure 7.11 goes approximately here, see next page ***

7.2.3 Revaluing and re-indexing an asset, CC/CPP Formula Accounting

There must be times when a firm decides that its initial choice of price index was wrong, and that a new asset value and a new index are required. Suppose, for instance, that the inventory from Figure 7.9 proved to be undesirable for some reason, and that some sort of write-down expense that explicitly reduced CCIFCO was considered preferable to allowing the drop in value to be reported as a holding loss via line (7.9.5). Figure 7.12 shows FAJEs on day $d_2$, that recognise an explicit write-down expense, revalue the inventory down to $c$, and base future revaluation on a new specific index $s$.

*** Figure 7.12 goes approximately here, see next page ***

Entries (7.12.1) through (7.12.8) can be compressed into three lines, but the present layout makes it easy to think through the steps. Lines (7.12.1) through (7.12.4) cancel (7.9.4) and (7.9.5). Lines (7.12.5) through (7.12.8) define the new
Trial Balance, 31 December

<table>
<thead>
<tr>
<th>Account</th>
<th>Dr</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank</td>
<td></td>
<td>204800</td>
</tr>
<tr>
<td>Accounts receivable</td>
<td>365000</td>
<td></td>
</tr>
<tr>
<td>Inventory</td>
<td>25s</td>
<td></td>
</tr>
<tr>
<td>Sales</td>
<td></td>
<td>3462.563g</td>
</tr>
<tr>
<td>Cost of sales</td>
<td>1918.718g</td>
<td></td>
</tr>
<tr>
<td>Holding gain/loss</td>
<td></td>
<td>25s+1918.718g+160200</td>
</tr>
</tbody>
</table>

Financial Accounting Reports at Year End

Income Statement, 1 January to 31 December

$  
Sales 3462.563*112 387807
less CoGS 1918.718*112 214896
CCIFCO 172911
plus Holding gain/loss 25*124 3100
-1543.845*112 -172911
+160200 160200 -9611
Comprehensive Income 163300

Balance Sheet, 31 December

<table>
<thead>
<tr>
<th>Account</th>
<th>Dr</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank</td>
<td></td>
<td>204800</td>
</tr>
<tr>
<td>Accounts receivable</td>
<td>365000</td>
<td></td>
</tr>
<tr>
<td>Inventory</td>
<td>3100</td>
<td></td>
</tr>
<tr>
<td>Owners' equity</td>
<td></td>
<td>163300</td>
</tr>
<tr>
<td></td>
<td>368100</td>
<td>368100</td>
</tr>
</tbody>
</table>

Figure 7.11 CC/CPP Reports based on 12 purchase and 365 sale transactions
Data: At day $d_2$, the firm in Figure 7.9 decides that inventory should be written down to $\$c$, and that a new specific price index, $s''$, is to apply in future. At that day, $d_2$, the previous specific index, $s$, has value $s_2$, the general index, $g$, has value $g_2$, and the new specific index, $s''$, has value $s''_2$.

Dr Inventory write-down expense  
  $a(s/s_0)$  
Cr Inventory  
  $a(s/s_0)$  
(7.12.1)  
(7.12.2)

Dr Holding gain/loss  
  $a(s/s_0 - g/g_0)$  
Cr Inventory write-down expense  
  $a(s/s_0 - g/g_0)$  
(7.12.3)  
(7.12.4)

Dr Inventory  
  $c(s''/s''_2)$  
Dr Inventory write-down expense  
  $c(s''/s''_2)$  
(7.12.5)  
(7.12.6)

Dr Inventory write-down expense  
  $c(s''/s''_2 - g/g_2)$  
Cr Holding gain/loss  
  $c(s''/s''_2 - g/g_2)$  
(7.12.7)  
(7.12.8)

After posting the four entries underlined above, the debit balance in Inventory write-down expense is $a(g/g_0) - c(g/g_0)$.

Figure 7.12: Revaluing and re-indexing an asset
asset value and holding gain. After posting lines (7.12.1) through (7.12.8), the balance in Inventory write-down expense is \( a(g/g_0) - c(g/g_2) \). On day \( d_2 \) this is just the purchase price for the inventory restated by the general level of prices less the value at \( d_2 \), i.e., \( $c \).

7.3 Valuing Fixed-Interest Liabilities at Market Yields, and the Use of Spreadsheets to Represent Formulae

*** Figure 7.13 goes approximately here, see next page ***

The final example for this chapter, presented in Figures 7.13 through 7.15, concerns discounting a fixed-interest security at market yields. Data for the example are given at the top of Figure 7.13. The journal entries in Figure 7.13 use formulae consisting of cell addresses from the spreadsheet in Figure 7.14. Figure 7.15 graphs some key cell values over the life of the bond. This example was deliberately chosen because it is difficult to model; it is not fair just to present the easy examples of Formula Accounting. It would have been possible to make the example more complex by adding a fourth variable, a general price index series, so that real interest expense (nominal interest expense less gains on holding a monetary liability) could be calculated. But the present example is probably quite complex enough.

Like conventional constant-journal-entry accounting, Formula Accounting for this bond example requires a journal entry to record the bond issue, then journal entries each 31st of December for profit determination, and each 23rd of March.
A firm sold a $1,000,000, 12% p.a. bond for $970,000 on 23 March, 1990 (the $30,000 difference being commission to the bank that handled the bond issue). Interest on the bond is payable annually for 5 years. The firm's financial year ends on 31 December. A schedule of market yields for bonds of similar risk and duration is as follows (use linear interpolation for intermediate dates):

<table>
<thead>
<tr>
<th>Date</th>
<th>Yield (p.a.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonds issued</td>
<td>23 March, 1990</td>
</tr>
<tr>
<td>1st year end</td>
<td>31 December, 1990</td>
</tr>
<tr>
<td>1st interest paid</td>
<td>23 March, 1991</td>
</tr>
<tr>
<td>2nd year end</td>
<td>31 December, 1991</td>
</tr>
<tr>
<td>Today</td>
<td>22 March, 1992</td>
</tr>
</tbody>
</table>

**Formula Accounting Journal Entries**

**Bond Issue, 23 March, 1990 (assume this is journal entry number 123456)**

<table>
<thead>
<tr>
<th>Description</th>
<th>Journal Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr Bank</td>
<td>E13 - E14</td>
</tr>
<tr>
<td>Dr Bond Issuing Expense</td>
<td>E14</td>
</tr>
<tr>
<td>Dr Bond Interest Expense</td>
<td>G33 + @sum(Adj1)</td>
</tr>
<tr>
<td>Cr Bond Yield Gain/Loss</td>
<td>G29 + @sum(Adj2)</td>
</tr>
<tr>
<td>Cr Bond Liability (Current)</td>
<td>G23 + @sum(Adj1) - @sum(Adj2)</td>
</tr>
<tr>
<td>Cr Bond Liability (Non-Current)</td>
<td>G24</td>
</tr>
</tbody>
</table>

Comment: See Figure 7.14 for details of J/E 123456's local spreadsheet. Adj1 and Adj2 are range names for adjustments on spreadsheet rows 52, 53, and 54 that compensate for changes in G23, G29, and G33 every 23rd of March, when interest payments are made. Initially all adjustments are zero.

**End of financial year, 1 January, 1991**

<table>
<thead>
<tr>
<th>Description</th>
<th>Journal Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr Profit and Loss</td>
<td>99,892 - 68,624</td>
</tr>
<tr>
<td>Dr Bond Yield Gain/Loss</td>
<td>68,624</td>
</tr>
<tr>
<td>Cr Bond Interest Expense</td>
<td>99,892</td>
</tr>
</tbody>
</table>

Comment: The figures in lines (7.13.8) and (7.13.9) cancel balances in their respective accounts on 1 January, 1991. They are not shown in Figure 7.14, which shows balances on 22 March, 1992.

**First interest payment, 23 March, 1991**

<table>
<thead>
<tr>
<th>Description</th>
<th>Journal Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr Bond Liability (Current)</td>
<td>&lt;&lt;123456&gt;&gt;C43</td>
</tr>
<tr>
<td>Cr Bank</td>
<td>&lt;&lt;123456&gt;&gt;C43</td>
</tr>
</tbody>
</table>

Comment: The double angle brackets in <<123456>>C43 is standard Lotus 123 notation for reference to another spreadsheet, in this case to cell C43 in the spreadsheet local to journal entry 123456.

Journal entries similar to (7.13.7) through (7.13.11) will be made in each subsequent year.

**Figure 7.13: Valuing a fixed-interest liability using market yields**

7 - 25(a)
to record the interest payments of $120,000. Because there are three "variables" to be considered, i.e., the discount rate is variable, the number of future cash flows decreases over time, and there must be some mechanism to account for possible default on interest payments, it is difficult to express ledger account balances as single formulae. However, as shown in Figure 7.14, it is possible to place all the relevant figures in a spreadsheet and to calculate the ledger account balances using a number of steps. Thinking along these lines leads to an important generalisation of the notion of a formula in Formula Accounting: a formula may be a call to any program that returns a number. In particular, it will be convenient if each Formula Accounting journal entry is automatically associated with its own spreadsheet, called its 'local' spreadsheet.

Figure 7.14 is thus the local spreadsheet for journal entry 123456, the first journal entry in Figure 7.13. It is assumed that the Formula Accounting ledger system provides full support for spreadsheets referenced in FA journal entries. The FA system needs to be able to refer to these spreadsheets when calculating ledger account balances.

*** Figure 7.14 goes approximately here, see next page ***

The spreadsheet in Figure 7.14, which has been crammed up to fit as much as possible onto a single page, is organized into four sections as follows:

(a) The SIMULATION section was included only to provide illustrative data for this chapter; it would not be required in practice. In this case a date,
### Fixed-Anual-Interest Bond Valuation Template

**INSTRUCTIONS:** Enter details of bond and year end in DATA area, E13..E21

**Record details of interest payments in row 43 WHEN PAID.**

--- ASSUMED INTEREST RATES Z p.a.---

**Today:**
- **day of mo:** 22
- **month:** 3
- **year:** 92

**Start Loan:**
- **day of mo:** 23
- **month:** 3
- **year:** 90

**End First Year:**
- **day of mo:** 01
- **month:** 1
- **year:** 91

**Pay 1st Interest:**
- **day of mo:** 23
- **month:** 3
- **year:** 91

**End Second Year:**
- **day of mo:** 01
- **month:** 1
- **year:** 92

**Pay 2nd Interest:**
- **day of mo:** 23
- **month:** 3
- **year:** 92

**Calculated Market:**
- **End Loan:** 23-Mar-95

**Yield for today:** 12.01219

---

**DATA:**
- **loan principal:** 1000000
- **commission paid to bank:** 30000
- **nominal yield (Z p.a.):** 12
- **date issued:** day of mo 23, month 3, year 90
- **years to maturity:** 5
- **financial yearend day of mo:** 31, month 12

---

**RESULTS:**
- **Bond Liability - Current:** 119962.74
- **Bond Liability - Non-Current:** 999605.48
- **Legal liability (loan principal from DATA):** 1000000.00

---

**WORKINGS:**
- **1 + daily market yield:** 1.00031062
- **Days since last interest payment:** 365
- **Bond duration (years):** 2.40289161

---

**Adjustments:**
- **Adjustment 1 (Dr):** 134822
- **Adjustment 2 (Cr):** -14822
- **Adj1 - Adj2 (Cr):** -120000

---

**Figure 7.14: Local spreadsheet for Journal Entry Number 123456**

---

**7-26(a)**
22 March, 1992, has been entered as Today\textsuperscript{9}, and the interest rates from Figure 7.13 have been entered in column H. From these data, the market yield for Today has been calculated in cell C11 as 12.01219.

(b) The DATA section allows details of the bond principal, nominal yield, date issued, years to maturity, etc., to be specified.

(c) In the WORKINGS section (at the bottom) the spreadsheet calculates scheduled interest payments (row 41), subtracts payments that have already been made (row 43), discounts the remaining payments (row 45) by the daily market yield (cell G35 = (1+C11/100)^{(1/365.25)})\footnote{In practice, Today would be calculated as in previous examples using some function like @int(@now).}, and places the discounted figures in row 46.

(d) Finally, in the RESULTS section Today's value for the bond (cell G26) is just the sum of present values from row 46. From this, the effective loan Principal on the day of the last interest payment is calculated (cell G28 = G26/G35^G36). For the data in the spreadsheet the next interest payment falls due tomorrow, day 366 of the (leap) year. Applying today's market interest rate of 12.01219% p.a. to the Principal in cell G28 for 365 days gives the liability of $1,119,568.22 shown in cell G26. Tomorrow, when interest is payable, the liability should have risen to $1,120,000 (there will be a small error because the compound interest formula used in the spreadsheet treats all years as having 365.25 days).
Assuming that interest is paid on the 23rd of March each year, Figure 7.15 shows how the Present Value of the liability (line A) and the Principal (line B) vary over the life of the bond. Saw-toothed line A shows the Present Value of the bond rising as interest payment dates get closer, then falling by the interest payment ($120,000) every payment day (23 March). In the first two years of the bond's life, market yields are higher than the nominal yield, so the Principal (line B) drops below $1,000,000. This is accounted for as a Bond Yield Gain. In the last three years it is assumed that the market yield equals the nominal yield. Discounting at nominal yields is sometimes claimed to be the normal historical cost approach to bond valuation. Thus the graph illustrates both market-yield and historical-cost bond valuation.

The vertical distance between lines A and B is the interest that would be payable if a bond with the Principal given in cell G28 (line B) had been issued last 23rd of March at current market yields. This Notional Interest Payable, calculated in cell G33, and graphed in Figure 7.15 as line C, drops to zero every 23rd of March when interest is paid. Finally, line D in Figure 7.15 is Interest Expense. It differs from Notional Interest Payable because it is closed to Profit and Loss at the end of each financial year. Because interest payment days will rarely fall due at financial year end, most interest payable/interest expense graphs will exhibit lagged saw-tooth patterns like line C with respect to D in Figure 7.15.

Returning to the journal entries in Figure 7.13, the first journal entry, lines
Figure 7.15: Present Value of the bond over its five-year life, and Related Time-Series.
(7.13.1) through (7.13.6), records the sale of the bond. Adj1 and Adj2 are range names for interest-payment-day adjustments stored in rows 52 and 53 of the spreadsheet (discussed in more detail below). Initially, all adjustment cells contain zeros. Treating debits as plus and credits as minus, the sum of all formulae in lines (7.13.1) through (7.13.6) is shown in the top line of the following box. The simplification in the next few lines of the box uses conventional rules of arithmetic (e.g., +x -x = 0) and knowledge of the formulae in the spreadsheet.

| E13-E14+E14+G33+@sum(Adj1)-G29-@sum(Adj2)-G23-@sum(Adj1)+@sum(Adj2)-G24 |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| =E13            | +G33            | -G29            | -G23            | -G24            |
| =E13            | +(G26-G28)      | -(E13-G28)      |                  | -G26            |
| =zero.          |                 |                  |                  |                  |

Thus as time passes the numbers in these cells may change, but journal entry 123456 will stay balanced. It is assumed that the Formula Accounting system can use its knowledge of the rules of arithmetic and of spreadsheets to check the balance of any journal entry in a similar fashion.

The second journal entry in Figure 7.13 is entered at the end of financial year when expense and gain/loss accounts are closed to Owners' Equity accounts (here assumed to be Profit and Loss). The figures in lines (7.13.8) and (7.13.9) are the balances reported in the Bond Yield Gain/Loss and Bond Interest Expense accounts at, say, 00:01 a.m. on 1 January, 1991, before this journal
entry is made\textsuperscript{10}. By 1 January, 1991, the market yield has risen to 14\% p.a., so the Gain on reduction in the Principal partly offsets the Interest Expense for the period. After the journal entry is posted, the debit balances in these accounts will be:

\begin{align*}
\text{Balance, Bond Yield Gain/Loss} & \quad -<<123456>>[G29+@\text{sum(Adj2)}] + 68,624 \\
& \quad (\text{line}) \quad (7.13.4) \quad (7.13.8) \\
\text{Balance, Bond Interest Expense} & \quad +<<123456>>[G33+@\text{sum(Adj1)}] - 99,892 \\
& \quad (\text{line}) \quad (7.13.3) \quad (7.13.9)
\end{align*}

where $<<123456>>[...]$ is to be interpreted to mean that the cell references in the square brackets belong to the spreadsheet local to J/E 123456. At this stage all adjustment cells are zero, so unless bond yield rates change, both these account balances will display as zero throughout 1 January, 1991.

The third journal entry in Figure 7.13 is to be made on 23 March, 1991, when the first interest payment of $120,000 is made. Journal entry lines (7.13.10) and (7.13.11) record this payment. Since the amount of money involved is a known constant, the debit and credit could be recorded as Dr 120,000, Cr 120,000. However, spreadsheet 123456 only works if details of payments are entered in row 43 as the payments are made. Thus entry of the formulae in lines (7.13.10) and (7.13.11) as references to spreadsheet 123456 ensures consistency between this journal entry and the spreadsheet that values the bond liability. Details of

\textsuperscript{10} They were calculated by setting the simulated date in cells C6..C8 to 1/1/91. Cell G29 then displayed $68,623.83, and cell G33 displayed $99,891.65.
a Lotus 123 macro written to record interest payments, and to update Adj1 and Adj2 in rows 52 and 53 of the spreadsheet, are given in the Appendix. The adjustments hold Bond Interest Expense, Bond Yield Gain/Loss, and Bond Liability (Current) unchanged on days when interest payments are made. Once journal entry line (7.13.10) has been posted to the ledger, the balance in the Bond Liability (Current) account will be

\[
\text{Credit balance: } \quad G23 + \@\text{sum(Adj1)} - \@\text{sum(Adj2)} - \langle<123456\rangle\rangle C43
\]

(line) (7.13.5) (7.13.10)

But the spreadsheet ensures (line 54) that the last three terms add to zero, so the balance displayed in the Bond Liability (Current) account will be just the figure in G23. As it should be, this is the present value of next year's interest payment.

Summarising, this final example of Formula Accounting shows two things. First, it shows a way that Formula Accounting could be used to value complex monetary liabilities as functions of time, changing market interest rates\(^{11}\), and payment of interest events. Second, it introduces the notion of a formula that invokes a program to calculate its value. The concept of a local spreadsheet, as illustrated, seems likely to be a particularly convenient way to implement many formulae.

\(^{11}\) Dependence on market interest rates can be "switched off" merely be setting cell C11 in Figure 8 to equal the figure in cell E15.
7.4 Comparing Formula Accounting and Constant-Journal-Entry Accounting

The good thing about Formula Accounting is that conventional Constant-Journal-Entry accounting is just a special case of Formula Accounting, i.e., Constant-Journal-Entry accounting is Formula Accounting with all formulae restricted to constants. This means that software designed for FA will work, without modification, with conventional Constant-Journal-Entry accounting. It also means that FA imposes no restrictions on the accountant: any real-world phenomena that can be recorded in a conventional accounting system can also be recorded in a Formula Accounting system.

Another desirable quality of Formula Accounting is that it enables time-varying asset and liability values to be modelled with one journal entry rather than many. Most firms have some assets or liabilities whose values may be expressed as functions of time or other independent variables (e.g., market prices, price index series, or even other account balances). In these cases, one FA journal entry can model asset or liability values as they change over time, whilst Constant-Journal-Entry accounting has to issue repeated journal entries, to try to keep up.

A more subtle difference between Formula Accounting and conventional Constant-Journal-Entry accounting is that Formula Accounting has the potential to bring a uniformity to asset and liability accounting that has hitherto not been possible. Constant-Journal-Entry accounting has always been a cumbersome tool for modelling continuous changes in value, but accountants have developed so many standard procedures to accommodate this limitation that the limitation
itself is hardly noticed. Table 7.1 contrasts some conventional constant-journal-entry accounting treatments of time-varying assets and liabilities with those possible under Formula Accounting.

*** Table 7.1 goes approximately here ***

Finally, because of this uniformity, it may be easier to teach Formula Accounting to novices than it is to teach conventional constant-journal-entry accounting. Consider two events: (1) payment of $1,000 for an computer, and (2) payment of $1,000 for an rights to occupy a building, i.e., a rental prepayment. Both resources acquired are accounted for as debits, but the former is usually treated as an asset, a 'good' thing, and the latter is usually treated as an expense, a 'bad' thing. The novice must ask 'Why are these similar events accounted for differently?' 'Why isn't the rental payment recorded as an asset too?' Formula Accounting makes it clear that in all exchanges of one economic resource for another, the resource acquired is an asset. Simultaneously, it records the way that asset's service potential is expected to be consumed, i.e., converted into an expense. The complex set of rules in the left-hand column of Table 7.1 need no longer be learnt.

Moreover, it could be argued that students are likely to learn accounting more readily if the accounting treatment of a situation parallels their intuitive understanding of the situation. If so, the first journal entry in Figure 7.1 could be easier to teach than the conventional constant-journal-entry accounting treatment. Students know from their own experience that once you buy a something its value normally changes over its life (land up, cars down, etc.).
### Table 7.1: Comparison of Techniques used for Modelling Changing Values in Conventional Constant-Journal-Entry Accounting and Formula Accounting

<table>
<thead>
<tr>
<th>Constant-Journal-Entry Accounting</th>
<th>Formula Accounting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track changing values of long-lived assets and interest-bearing liabilities using standing journal entries for depreciation, amortisation, and interest expense.</td>
<td>When an asset is acquired, use formulae like those in Figure 7.1. When a fixed-interest liability is incurred, use formulae like those in Figure 7.13.</td>
</tr>
<tr>
<td>Classify assets as expenses if they are expected to be short-lived, e.g., prepaid rent is usually treated as an expense because it is not worth the effort required to treat it as a diminishing-value asset.</td>
<td>Record long and short-lived assets the same way: as assets. Only their lives are different.</td>
</tr>
<tr>
<td>Reject proposals for valuing well-defined classes of assets and liabilities (for which market prices are readily obtainable) at current values, on the grounds that it is too much work to implement such values in the accounts.</td>
<td>Build telecommunications links to stock markets, foreign exchange markets, and commodities markets so that changes in prices are reflected automatically in the accounts.</td>
</tr>
</tbody>
</table>
Formula Accounting mirrors this experience. Of course, it could equally well be argued that the use of variables in Formula Accounting makes it harder for less numerate students (those who dropped Mathematics at school) to understand. Resolution of such conflicting hypotheses is beyond the scope of this thesis.

7.5 Implementation of Formula Accounting in today's General Ledger software, i.e., in a system without an Interpreter

In the next chapter, Chapter 8, a database schema for Formula Accounting is developed primarily to receive journal entries generated by the Interpreter. However, Formula Accounting could be included in any conventional accounting system, quite independently of the use of an Interpreter to generate the journal entries. This section briefly considers how Formula Accounting might be used in a situation where no Interpreter exists, and data are funneled straight from various feeder subsystems into a FA general ledger.

Adapting existing general ledger software to Formula Accounting does not appear to present any intractable problems. Each account must contain the formula representing the current account balance, instead of a constant number. This formula might refer directly to some time-varying independent variables, or be a reference to a cell in a spreadsheet, or a call to a program. Since the FA system should be able to calculate account balances at any time, it will need to control modification and deletion of spreadsheets and/or programs used to calculate account balances. The FA system also needs to be able to simplify account balance formulae whenever new journal entries are posted to an account, and to ensure that journal entries always balance. There are some
challenges in this list of requirements, but none is impossible.

Users may want to store a number of years of actual historical monthly balances in each account. For instance, large computer general ledger software packages from vendors like McCormack and Dodge [1983] and MSA [1986] provide for some years of historical monthly balances to be stored in each account record. In such systems, storage of monthly balances is already part of the normal monthly reporting cycle. With a FA general ledger all that the program would need to do is to intercept the account balance information after it has been evaluated for reporting purposes, and store it as a number (not a formula) in the appropriate field of the account record. There would be virtually no other change to the existing general ledger program. The budgeting component of the general ledger system could remain as it is, although users would probably press for formulae-based budgeting capabilities too, e.g., for flexible budgets.

If a conventional general ledger system were modified to accommodate Formula Accounting it would probably be desirable to build some sort of primitive Interpreter to simplify generation of FA journal entries. Because of unfamiliarity with Formula Accounting it can take some time to 'think through' journal entries like those in this chapter. For journal entries that are keyed-in by hand, it would therefore be helpful to have some reasonably intelligent 'front end' program that could suggest the formulae for different event types and ensure (as present-day systems do) that journal entries balance. The journal entries in Sections 7.1 through 7.3 of this chapter provide good examples of the sort of journal entry templates that a 'front end' program might use. The 'front end' could also be asked to check that journal entries balance over time, i.e.,
that the sum of the coefficients of each independent variable used in the journal entry exactly balance. Finally, all the existing feeder systems that supply journal entries to the existing general ledger system would have to be modified to generate FA journal entries.

7.6 Chapter Summary

Although centuries of use of conventional constant-value journal entries have proven them to be a simple, robust method for capturing some of the essential characteristics of economic events, it is difficult to use them to model asset and liability values that change continuously over time. With constant-value-journal-entry accounting one is forced to issue repeated journal entries, e.g., standing journal entries, to try to reflect changing values in ledger account balances.

Formula Accounting offers accountants the chance to model time-varying asset and liability values much more simply and directly. In Formula Accounting, ledger account balances are stored as spreadsheet-style formulae. Whenever an account balance is displayed on a computer screen, or in a report, the computer simply evaluates the formula and displays the number, as it does with a spreadsheet. If formulae are defined in terms of time-varying variables, ledger balances will change automatically as the underlying variables change.

To demonstrate how Formula Accounting works, this chapter has presented a series of five examples using different combinations of independent variables and different types of inflation/current value accounting. In Figure 7.1 the
independent variable is time; in Figure 7.4 it is a market price; in Figure 7.5 it is another account balance. Figure 7.6 uses a combination of price indices and time as independent variables to implement an inflation accounting system. Figures 7.7 through 7.12 showed CC/CPP accounts for fixed assets and trading accounts. Figure 7.13 uses a combination of market interest rates and time for valuing long-term liabilities. Figure 7.13 also illustrates the advantages of including spreadsheets as an integral part of a Formula Accounting system -- they make it easy to define formulae of seemingly arbitrary complexity.

In Section 7.4 it was shown that a decision to convert from a conventional constant-value ledger system to Formula Accounting would involve very little risk for either software vendors or users. This is because conventional constant-value journal entries will work, without modification, in a Formula Accounting general ledger system.

On a more practical level, Section 7.5 considered the implementation of FA in an existing general ledger system. The main changes are that Formula Accounting systems must be able to store ledger balances as formulae, to post journal formulae to existing ledger balance formulae, and to manage local spreadsheets. In such systems it would seem desirable to have some sort of primitive Interpreter as a 'front end', both to shield the user from more complex formulae, and to ensure that new journal entries balance over time.
Appendix 7.1: Capturing the Adjustments called Adj1 and Adj2 in the Spreadsheet in Figure 7.14

For the spreadsheet in Figure 7.14 it was necessary to write a macro to keep the ledger in balance when interest payments were recorded in row 43. The problem is that when the interest payment is recorded in row 43 the figures in cells G23, G29, and G33 all change, yet the ledger account balances that depend on these cells should not change. For instance, by journal entry line (7.13.3), Interest Expense depends on cell G33. Yet Notional Interest Expense (line D in Figure 7.15) should not change when the interest payment is made (line C in Figure 7.15). The solution employed in the Figure 7.14 spreadsheet was to write a macro for recording interest payments. The macro works as follows:

Adj1 and Adj2 are range names for cells C52..G52 and C53..G53 in the spreadsheet, initially all zero. When an interest payment is recorded in row 43 the values in cells G29 and G33 change. Exactly opposite changes are then stored in Adj1 and Adj2 so that balances calculated by lines (7.13.3) through (7.13.5) are unchanged. The two changes always net out to exactly the amount of the interest payment, in this case to $120,000 (see line 54 of the spreadsheet).

For example, in March 1991 when the interest payment of $120,000 is recorded in cell C43, cell G33 drops from 134,822 to zero, and cell G29 rises by 14,822. The macro then stores +134,822 in cell C52, and -14,822 in cell B53. In general, whenever interest payments are made, changes in cells G33 and G29 are captured in rows 52 and 53. The formulae in lines (7.13.3) and (7.13.4) add these changes back so that the interest payments have no impact on the corresponding expense and gain accounts.
Chapter 8: A Prototype Formula Accounting System

This chapter sets the stage for the next two chapters. As shown in the top lines of Figure 8.1, Chapter 9 introduces the Resources and Exchange Events (REE) data model and Chapter 10 outlines the rules used by the Interpreter as it generates Formula Accounting journal entries (FAJEs). There are two reasons for including this rather technical chapter at this stage of the thesis. First, Chapter 9 uses the three data modelling tools introduced in this chapter:

(i) NIAM data modelling diagrams,
(ii) SQL descriptions of relational database tables, and
(iii) Prolog as a language for implementing and manipulating relational tables.

![Figure 8.1 Relationship between Chapters 7 through 10](image)
NIAM diagrams are used in preference to, say, the Entity-Relationship (E-R) diagrams used by McCarthy [1979], because they are supposed to ensure that any relational database derived from them is in what Nijssen and Halpin call "optimal normal form". In the following passage Nijssen and Halpin [1989, p.254] describe optimal normal form (ONF) as similar to fifth normal form [Kent, 1983]:

Starting with a conceptual schema, the ONF algorithm thus produces a table design satisfying three criteria: (1) no repeating attributes; (2) no redundancy or update problems; (3) minimum number of tables satisfying (1) and (2). ... The ONF algorithm typically generates tables in fifth normal form (5NF); moreover, these are optimized further since the number of 5NF tables in the overall schema has been minimized. For this reason the term "optimal normal form" is used.

Thus by starting the database design with a NIAM diagram, one can be confident that there is little chance of update anomalies in the resultant relational database tables.

A second, equally important, reason for this chapter is that in reading the discussion of the REE data model in Chapter 9, which is quite abstract, it is helpful to have in mind a clear, "concrete" example of a Formula Accounting system. Knowledge of the information required for the target Formula Accounting model is used to guide choices about which facts about the real world should, or should not, be modelled in REE. For both these reasons it was decided that this short chapter, Chapter 8, should present details of the prototype Formula Accounting system constructed for testing the Interpreter discussed in Chapter 10.

Aside: The use of the word "model" in data modelling is sometimes confusing. Some models are models of the real world, others are actually methodologies. For
instance, the REE model described in the next chapter is a model of the very small part of the real world that is relevant to accountants. However, when people speak of the Relational Model [Codd, 1970], or the Entity-Relationship Model [Chen, 1976; Teorey, Yang, and Fry, 1986], they are talking of languages, or methodologies, used for model building. Modelling languages define how things may be said, and what may be said, in building models of some other real-world Universe of Discourse. In this thesis methodologies are spelt with an upper-case "M" -- they are Models.

8.1 A NIAM Data Model for Formula Accounting

Like any general ledger system, a FA system must be able to store journal entries, details of ledger accounts, reporting sequences, and relationships between accounts. Storing ledger balances is optional, since they can be derived from the history of journal entries, but storing balances saves having to re-scan large numbers of transactions each time an account balance is required. Figures 8.2 and 8.3 present a NIAM conceptual schema diagram [Verheijen and Van Bekkum, 1982; Nijssen and Halpin, 1989] for a Formula Accounting system.

*** Figures 8.2 and 8.3 go approximately here ***

Since NIAM has not been widely used in the Accounting literature, a few words of explanation of the diagram may be helpful. The circles in Figure 8.2 are what NIAM calls object types. Object types are things of interest in the data model. In

---

1 NIAM is also a methodology: Nijssen's Information Analysis Methodology.
Roles (predicates): e.g., r1: "Balance x is balance of Account y"  
r1": "Account y has Balance x"  
r2: is class of  
r3: is provision/expense a/c for  
r4: is account for  
r5: has  
r6: has debit/credit  
r7: is date of  
r8: describes  
r9: belongs to  
r10: is controlled by
Figure 8.3: (Incomplete) Subtype Hierarchy for the Object type "Account"
Figure 8.2 "Account" and "Balance" are both object types. Facts about objects of interest are stored in tables (with as few columns as possible) in a NIAM database. 

Fact types in NIAM are depicted by rectangles showing the number of columns in the corresponding table. For instance, the fact "Account 123 has Balance $456.78" would be stored in the two-column table linking the object types Account and Balance. In a NIAM diagram, key fields (which uniquely identify rows in the table) are underlined beneath the appropriate column of the rectangle representing the table. In Figure 8.2, each Account may have one and only one Balance, whereas the same balance, e.g., zero, might apply to a number of accounts, so the column nearest Account has been underlined.

To illustrate the meaning of Figure 8.2, consider storing the journal entry in Figure 8.4:

\[
\begin{array}{|l|l|l|l|}
\hline
JE1 & 900101 & 1 & Dr & Inventory 1 & 200^s \\
2 & Cr & Holding Gain/loss & 200^*(s-g) \\
3 & Cr & Cash at Bank & 20000 \\
4 & Cr & Holding Gain/loss & 200^g-20000 \\
\hline
\end{array}
\]

being purchase of 40 units of Inventory 1 for $20,000 cash, using CC/CPP accounting with both index series s and g having value 100 on date 900101.

**Figure 8.4: A Sample FA journal entry (the January purchase in Figure 7.10)**

From a schema design viewpoint the features of this journal entry are, first, that it consists of a number of journal entry lines, where each line is a debit or credit
to one, and only one, account, and second, that each journal entry has some "header" information, like the Journal Entry number (JE1), date, and narration, that occurs only once per journal entry. To record the journal entry in the NIAM database one would proceed as follows:

First, the journal entry date would be recorded by storing in the table between Date and Journal_Entry_Header a line with "900101" on the Date side, and "JE1" on the Journal_Entry_Headerside. The narration would be recorded by storing in the table between Narration and Journal_Entry_Header a line with "being purchase of 40 units of Inventory 1 ..." in the Narration side, and "JE1" in the Journal_Entry_Header side.

Second, details of each journal entry line would be recorded in the appropriate tables. For instance, to record the first journal entry line a new line would be added to the table between Journal_Entry_Header and Journal_Entry_Detail. The new line would contain "JE1" in the Journal_Entry_Number column (the Header side of the table), and "JE1,1" in the Journal_Entry_Line column (the Detail side of the table)\(^3\). Then, (i) in the table between Journal_Entry_Line and Account a new line would be added with "Inventory 1" in the Account column, and "JE1,1" in the Journal_Entry_Line column; (ii) in the table between Journal_Entry_Line and Amount a new line would be added with "JE1,1" in the Journal_Entry_Line column, and "+200*s" in the Amount column; and (iii) in the table between Account and Balance the line where the Account column is "Inventory 1" would be identified, and the balance in the Balance column updated. An identical series of steps would

---

\(^3\) This redundancy, where "JE1" is stored in both columns of the table, is eliminated later in the ONF procedure.
be followed for each of the other four journal entry lines, using line numbers 2 through 5.

As should now be clear, a database described by a NIAM conceptual schema captures "elementary facts", as Nijssen calls them, in a laboriously simple manner. The database consists of many tables (often with only two columns), with each elementary fact stored as a row in one of these tables. Some elementary facts require more than two columns to describe them. For instance, a price index series requires a ternary fact type that says: "series x had value y on date z". This sentence about x, y and z cannot be broken into binary facts without losing meaning. So the NIAM table to store facts of this type has three columns, not two.

In addition to its "dumbbell" diagrams\(^4\), NIAM also uses subtype hierarchies for classifying object types. Figure 8.3 is an (incomplete) tree data structure defining the way sub-types of the object type "Account" are related. Similar diagrams have appeared regularly in the accounting literature, e.g., Eaves [1966, p.428], Lieberman and Whinston [1975, p.252], and Everest and Weber [1977, p.343], to name but three papers, but Figure 8.3 is more than a generalization hierarchy: it also contains sequencing information for accounting reports. When accounting reports are generated, information is expected to be grouped so that all assets, all liabilities, etc. are together. The tree in Figure 8.3 contains this ordering information, as well.

---

\(^4\) For binary relationships, the circles linked by rectangles look like dumbbells.
Not shown in either Figure 8.2 or 8.3 is the representation of journal-entry-line formulae and account-balance formulae. The particular implementation of formulae used for the prototype system is discussed in Section 8.4 below.

8.2 A Relational Model for Formula Accounting

*** Figure 8.5 goes approximately here, see next page ***

Using the optimal normal form (ONF) algorithm described by Nijssen and Halpin [1989, Ch.11] the schema in Figure 8.2 was converted into the relational database form defined in Figure 8.5. (In essence, the ONF algorithm says that wherever possible tables with common keys should be joined on their common key.) This relational database schema is supposed to be in fifth normal form [Kent, 1983]. It is so simple that there is no risk of update anomalies.

Figure 8.5 is similar to Everest and Weber's intuitively-developed, second Financial Accounting model [1977, p.356]. The first table definition in Figure 8.5 defines a table named GL_Account with four columns: GL_Account_No, Description, Class, and Balance_Formula. The remaining lines at the top of Figure 8.5 define five additional tables called Journal_Entry_Header, Journal_Entry_Detail, Account_Hierarchy, GL_Depreciation, and GL_Amortization, described in more detail below. Together, these six tables store the information required to build a conventional accounting system.

---

5 Weber has confirmed by letter that this model was developed intuitively.
GL_ACCOUNT (GL_ACCOUNT_NO, DESCRIPTION, CLASS, BALANCE_FORMULA)

JOURNAL_ENTRY_HEADER (JENQ, DATE, ENO, NARRATION)

JOURNAL_ENTRY_LINE (JENO, LINE_NO, GL_ACCOUNT_NO, JEL_FORMULA)

ACCOUNT_HIERARCHY (PARENT_CLASS, CHILD_CLASS, CHILD_SEQUENCE)

GL_DEPRECIATION (ASSET_AT_COST_ACCOUNT, ACCUM_DEPR_ACCOUNT, DEPR_EXPENSE_ACCOUNT)

GL_AMORTIZATION (AMORT_ASSET_ACCOUNT, AMORT_EXPENSE_ACCOUNT)

Examples

(a) FAJEs from Figure 8.4

JOURNAL_ENTRY_HEADER (1, 1 January, ENO, 'Purchase of inventory for cash')

JOURNAL_ENTRY_LINE (1, 1, 11401, 200*s)
JOURNAL_ENTRY_LINE (1, 2, 31501, -200*(s-g))
JOURNAL_ENTRY_LINE (1, 3, 11102, -20000)
JOURNAL_ENTRY_LINE (1, 4, 31501, -200*g+20000)

GL_ACCOUNT (11102, 'Cash at Bank', 'Current Asset', -20000)
GL_ACCOUNT (11401, 'Inventory', 'Current Asset', 200*s)
GL_ACCOUNT (31501, 'Holding gain/loss', 'Owners Equity', 200s+200g-20000)

(b) The Account Hierarchy table

The Account_Hierarchy table is only used for reporting purposes. It defines the chart of accounts as a tree with a specific reporting sequence. Representing Figure 8.3 in this table, the first five rows would be as follows:

<table>
<thead>
<tr>
<th>Parent_Class</th>
<th>Child_Class</th>
<th>Child_Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account</td>
<td>Asset Account</td>
<td>1</td>
</tr>
<tr>
<td>Account</td>
<td>Liability Account</td>
<td>2</td>
</tr>
<tr>
<td>Account</td>
<td>Owners' Equity Account</td>
<td>3</td>
</tr>
<tr>
<td>Asset Account</td>
<td>Current Asset Account</td>
<td>1</td>
</tr>
<tr>
<td>Asset Account</td>
<td>Fixed Asset Account</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Child_Sequence shown above causes Assets to be reported before Liabilities, Current Assets before Fixed Assets, and so on.

Figure 8.5: Relational tables for Formula Accounting
Beneath the table declarations in Figure 8.5 there is a small example of data for the first three tables. These correspond to the journal entry in Figure 8.4.

The fourth table defined in Figure 8.5, ACCOUNT_HIERARCHY, is used to store the chart-of-accounts tree shown in Figure 8.3. The first five lines of this table are shown at the bottom of Figure 8.5. Note that the Class column in the GL_Account table contains pointers to leaf nodes in the Account_Hierarchy tree. For example, if individual Finished Goods Inventory accounts were maintained in the ledger system, their Class column could be defined as "Finished Goods Inventory Account", and this would match a line (a leaf node) in the ACCOUNT_HIERARCHY table which showed that "Finished Goods Inventory" was a child type of "Inventory", which was in turn, a child type of "Current Assets".

Finally, the last two tables of Figure 8.5, GL_DEPRECIATION and GL_AMORTIZATION, identify the accounts to be used when any asset is depreciated or amortized. The relationships between these accounts is implicit in most general ledger systems.

8.3 Implementing the FA Relational Database in Prolog

*** Figure 8.6 goes approximately here, see next page ***

The programming language used for building the prototype, Prolog, provides a simple mechanism for representing relational databases. It also provides a powerful data manipulation language. With the exception of the formulae, e.g., 200*s and 200*(s-g), the information in Figure 8.5 can be represented in Prolog.
jeh(1, 900101, Eno, 'Purchase of inventory for cash').

jed(1, 1, 11401, '200\times s').
jed(1, 2, 31501, '-200*(s-g)').
jed(1, 3, 11102, '-20000').
jed(1, 4, 31501, '-200*s+20000').

chart_of_accounts(11102, 'Cash at Bank', 'Current Asset', '-20000').
chart_of_accounts(11401, 'Inventory 1', 'Inventory', '200\times s').
chart_of_accounts(31501, 'Holding gain/loss', 'Owners Equity', '200\times s+200*g-20000').

Note: jeh corresponds to JOURNAL_ENTRY_HEADER of Figure 8.5,
jed corresponds to JOURNAL_ENTRY_LINE of Figure 8.5, and
chart_of_accounts corresponds to GL_ACCOUNT of Figure 8.5.

Figure 8.6 Representing Figure 8.5 data in Prolog
almost directly as it appears in that figure. For instance, Figure 8.6 is syntactically valid Prolog. Prolog treats terms beginning with an upper-case letter, e.g., Eno in the first line of Figure 8.6 as variables, so the table names must begin with a lower-case letter. Each of the eight lines in Figure 8.6 asserts a Prolog fact. A Prolog fact begins with a predicate (in relational database language, a table name), in this case jeh, jed, or chart_of_accounts, followed by the data for the "table" in brackets separated by commas. Character strings in Prolog are delimited by single quotes. Each Prolog fact is terminated by a full-stop (period).

8.4 Representing Formulae in the Prototype Formula Accounting System

Ideally, a Formula Accounting system should be able to add (post) formulae of arbitrary complexity. For instance, it was suggested in Chapter 7 that it would be convenient in a practical FA system if each formula could be represented by a spreadsheet or some even more complex program.

Now addition of formulae, particularly conditional formulae, is a non-trivial problem. Spreadsheet programs today do not add formulae. They simply evaluate formulae that someone has keyed into a cell, then perform logical tests or do arithmetic with the resultant numbers. However, it was clear that some sort of formula-adding mechanism was required for the prototype system.

*** Figure 8.7 goes approximately here, see next page ***

---

6 These predicate names, jeh, jed, and chart_of_accounts are the names used in the Prolog system discussed shortly. The names were used in the first implementation of the Interpreter in 1986 and as it would take considerable effort to change them they have not changed since then.
CHAPTER 1

Overview

This tutorial explains how to do simple algebraic calculations such as the following:

Example 1
Potpourri of Maple commands

\[ \sum_{i=1}^{n} i^2 \text{, and simplify the result. Maple commands typically end with a semi-colon.} \]
\[ \text{> sum(i^2, i=1..n);} \]
\[ 1/4 \ (n + 1) - 1/2 \ (n + 1) + 1/4 \ (n + 1) \]
\[ \text{> simplify(“);} \]
\[ 1/4 n + 1/2 n + 1/4 n \]

\[ \text{> expr1 := a*(x-1);} \]
\[ \text{Compute } \int a(x - 1) \cos(a(x - 1)dx). \]
\[ \text{> int(expr1*cos(expr1),x);} \]
\[ \cos(a(x - 1)) + a(x - 1) \sin(a(x - 1)) \]

\[ \text{> divide(x^11-x^10+x-1, expr1/a, ’quotient’);} \]
\[ \text{true} \]
\[ \text{> print(quotient);} \]
\[ x + 1 \]

It will also explain how to do more involved computations by using the Maple programming language, such as the following:
The problem of adding formulae comes under the general heading of "symbolic computation". Simon [1990] recently reviewed four software products, including the two most widely known, Mathematica and Maple, that perform symbolic computation. His review indicates that Maple is the fastest (in 13 of 15 test problems, with solutions coded by the vendors), and probably the most powerful, of the four products. Figure 8.7, a copy of the first page from the Maple tutorial [Char, Geddes, Gonnet, Monagan, and Watt, 1990], gives an indication of the power of these programs. The first example in Figure 8.7 computes and simplifies the formula for the sum of a series; the second integrates a complex expression; and the third calculates and prints the quotient from a polynomial division. It would be ideal if Maple's data structures and procedures were available for representing and manipulating formulae for Formula Accounting. And, of course, they could be, for a full, commercial implementation of Formula Accounting.

However, programs like Maple are self-contained, not readily accessible from Prolog, and were therefore not available for the prototype system. So, after reviewing a number of Computer Science "data structures" books on addition of polynomials, e.g., Knuth [1973], Horowitz and Sahni [1976], and Lewis and Smith [1976], it was decided that it would be sufficient for testing purposes if a formula could be conditional (if...then...else...) and that the basic arithmetic operations of addition, subtraction, multiplication, division, and exponentiation were supported.

For instance, a pseudocode formula for historical-cost straight-line depreciation of a fixed asset is as follows:
if (Date_now - Date_acquired) < Life_in_days
then Depreciation = (Date_now - Date_acquired) * Daily_rate
else Depreciation = Acquisition_cost

where Daily_rate =

(Acquisition_cost - Estimated_residual_value)/Expected_life_in_days.

This formula has three parts: the condition in the first line, the if-true part in the second line, and the if-false part in the third line. As a second example, the formula for reducing-balance depreciation is unconditional but it uses exponentiation:

Depreciation = Acquisition_cost * (1 - (Reduction_factor^Life_used))

where ^ means "raised to the power of",
Reduction_factor = Estimated_residual_value/Acquisition_cost, and
Life_used = ((Date_now - Date_acquired)/Expected_life_in_days)

After considering a number of potential formulae and data structures it was decided that formulae in the prototype system would be represented by combinations of two predicates (tables) as follows:

formula(Fno,LHS_UFno,Condition,RHS_UFno,True_UFno,False_UFno).
uncondit_formula(UFno,Coefficient,Multiplication_list).
In these predicate definitions, symbols are defined as follows: Fno stands for a conditional formula number, UFno stands for unconditional formula number (either LHS, RHS, True, or False), Condition is a test operator (such as =, <, =<, and so on), Coefficient is any constant, positive or negative, and Multiplication_list is a list of variables of arbitrary powers to be multiplied together. Addition of polynomial terms is represented by a series of uncondit_formulae all with the same unconditional formula number.

*** Table 8.1 goes approximately here, see next page ***

Table 8.1 provides some examples of the way formulae are represented using this tree-like data structure. In Row 1, the middle column contains an unconditional formula, \(-12+3x^2y^3-4z\). With exponentials represented as lists\(^7\), where \([x,2]\) means \(x\) raised to the power 2, this may be thought of as three terms:

\[
(-12)[] + (3*[x,2]*[y,3]) + (-4*[z,1]).
\]

The above unconditional formula is represented in the prototype system by the four Prolog facts in the right-hand column. Each line in the right-hand column of Table 8.1 is a Prolog fact. In the top line is a formula predicate where four of the six parameters (the left-hand side formula, the condition, the right-hand side formula, and the false value for the formula) are all null (represented in Prolog by the "_" symbol). This Prolog fact thus says that "the value of formula 1 is the value

\(^7\) Lists in Prolog are stored within square brackets. \([a,b,c]\) is a list containing three terms, \(a\), \(b\), and \(c\). \([]\) is an empty list. \([a,b],[c,d]\) is a list containing two terms, both of which are themselves lists.
<table>
<thead>
<tr>
<th>Row</th>
<th>Formula</th>
<th>Representation in the Prototype System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-12+3x^2y^3-4z</td>
<td>formula(1,-12,1) uncondit_formula(1,-12,[]). uncondit_formula(1,+3,[[x,2],[y,3]]). uncondit_formula(1,-4,[[z,1]]).</td>
</tr>
<tr>
<td>2</td>
<td>if 3x+2y^2z = 9z then -12+3x^2y^3-4z else 100+2x^0.5-0.06x3z(5+3y)</td>
<td>formula(2,2,=,3,4,5). uncondit_formula(2,+3,[[x,1]]). uncondit_formula(2,+2,[[y,2],[z,1]]). uncondit_formula(3,9,[[z,1]]). uncondit_formula(4,-12,[]). uncondit_formula(4,+3,[[x,2],[y,3]]). uncondit_formula(4,-4,[[z,1]]). uncondit_formula(5,100,[]). uncondit_formula(5,2,[[x,0.5]]). uncondit_formula(5,-0.06,[[x,3],[z,6]]). uncondit_formula(6,-5,[]). uncondit_formula(6,+3,[[y,1]]).</td>
</tr>
<tr>
<td>3</td>
<td>200s - 200g</td>
<td>formula(3,<em>,</em>,_,7,__). uncondit_formula(7,200,[[s,1]]). uncondit_formula(7,-200,[]).</td>
</tr>
<tr>
<td>4</td>
<td>The balance formula above after posting the following journal entry formula to it: 200g - 20000</td>
<td>formula(3,<em>,</em>,<em>,7,</em>). uncondit_formula(7,200,[[s,1]]). uncondit_formula(7,0,[[g,1]]). uncondit_formula(7,-20000,[]).</td>
</tr>
<tr>
<td>5</td>
<td>0.6 * balance in GL_ACno 'X'</td>
<td>formula(4,<em>,</em>,<em>,8,</em>). uncondit_formula(8,6,[[gl_account(X),1]]).</td>
</tr>
</tbody>
</table>

Table 8.1 Representation of formulae in the prototype system

8-12(a)
of \textit{uncondit\_formula\_1}. (It is just a coincidence that both the formula number and the uncondit\_formula number are the same, i.e., 1.)

Uncondit\_formula\_1 is represented by the combination of three Prolog facts in the next three lines (all three have unconditional formula number, UFno, equal to 1). These three Prolog facts correspond to the three additive terms of the polynomial formula in the middle column. Their values are \textit{added} together in evaluating \textit{uncondit\_formula\_1}. The first uncondit\_formula fact, i.e.,

\texttt{uncondit\_formula(1,-12,[]),}

has a constant coefficient, -12, and an empty Multiplication\_list, []. The evaluation routine treats the empty list as a 1, so \texttt{uncondit\_formula(1,-12,[])} represents -12*1, or simply -12. The second term in the polynomial in Row 1 is +3x^2y^3. The corresponding Prolog fact is

\texttt{uncondit\_formula(1,+3,[[x,2],[y,3]]).}

Here the coefficient is +3, the first term of the Multiplication\_list is [x,2], meaning x squared, and the second term in the Multiplication\_list is [y,3], meaning y cubed. Having [x,2] and [y,3] as terms in the one Multiplication\_list, i.e., [[x,2],[y,3]], means that their values are \textit{multiplied} when evaluating the unconditional formula. Finally, the third polynomial term in the formula in Row 1, Column 2 is -4z. This is represented by the third uncondit\_formula, with coefficient -4, and a Multiplication\_list with only one term, [z,1].

The more complex example in Row 2 of Table 8.1 is a conditional formula. The Prolog fact for formula number 2 in the right-hand column, i.e., formula(2,2,=,3,4,5), says "IF at the time of evaluation the value of uncondit\_formula\_2 equals ("equals" because the test operator is ";=") the value of
uncondit_formula 3, THEN the value of formula 2 is the value of uncondit_formula 4, ELSE it is the value of uncondit_formula 5. Uncondit_formula 2 is a polynomial with two terms, 3x and 2y²z. Uncondit_formula 3 is a polynomial with one term, 9z. Uncondit_formula 4 is a polynomial with three terms, identical to the polynomial in the first example. Uncondit_formula 5 is a formula with three terms: a constant, 100; a single-variable term, 2x⁰.⁵, i.e., 2 times the square root of x; and a third term where 0.06 is multiplied by x cubed and then by z raised to the power (-5 + 3y). This exponent, -5 + 3y, is represented by a separate uncondit_formula, in this case uncondit_formula 6, which itself requires two Prolog facts, one for the -5, and the other for the 3y.

In general:

1. Each Formula Accounting journal-entry formula is represented in the prototype system by one Prolog fact with predicate formula. For example, each Row of Table 8.1 has only one formula fact.

2. If the FA formula is unconditional its formula will refer to only one uncondit_formula. For example, in Row 1, Table 8.1, formula 1 refers only to uncondit_formula 1 (the other four uncondit_formula parameters in formula 1 are null (_)).

3. If the FA formula is conditional, its formula will refer to four uncondit_formula. For example, in Row 2, Table 8.1, formula 2 refers to uncondit_formula 2, 3, 4, and 5.
4. In each uncondit_formula addition, multiplication, and exponentiation are represented as follows:

(a) separate facts with the same uncondit_formula number are **added** (negative coefficients are used for subtraction), e.g., the terms in the three uncondit_formula facts in Row 1 are added.

(b) adjacent terms in a list within any one fact are **multiplied** (negative exponents in the Multiplication_list are used for division), e.g., the list $[[x,1],[y,1],[z,-1]]$ means $x^*y/z$;

(c) the exponent of each variable is the second item in each sub-list, e.g., the sub-list $[x,2]$ represents $x$ squared.

In the data structure above, it would have been possible to allow a **formula** to contain references to other formulae. This would enable compound and nested conditions (e.g., IF (a = b) AND (c > d)... or IF a = b THEN (IF c > d... )) to be used in FA formulae. However, the prototype system used in this thesis does not allow such compound conditional formulae. This restriction was imposed merely to simplify comparison of conditional formulae when posting journal entries. For instance, the following two conditions have equivalent meaning:

$$(a < b) \text{ OR } ((a = c) \text{ AND } (c =< b))$$

$$(a =< b)$$
(and much nastier examples could be constructed). But it requires much extra coding effort to check whether compound conditions are logically equivalent. Clearly, a data structure like that used in spreadsheets, where conditional formulae in one cell can refer to conditional formulae in other cells would be more general and more powerful than the system proposed here. However, as noted earlier, spreadsheets work only with the results of evaluating their formulae. They do not reason about the equivalence or otherwise of their formulae, so the reasoning required for comparing conditional formulae in Formula Accounting is considerably more complex than in a spreadsheet. To keep the prototype FA system simple, compound and nested conditions were omitted.

Completing the examples in Table 8.1, Row 3 is the simple unconditional formula debited to Holding gain/loss account 31501 in the second jed line of Figure 8.6 above. It shows the balance in account 31501 BEFORE the fourth jed line in Figure 8.6, i.e., \(-200^*g + 20000\), is posted. Row 4 shows the balance formula AFTER posting (adding). Row 4 is determined by adding coefficients of the two formulae where the Polynomial_term lists match exactly (list \([g,1]\) in this case), or by adding an extra uncondition formula fact where there is no match \((-20000,[]\) in this case). If the formula is conditional, the conditions and the Polynomial_term lists must both match before coefficients may be added. Row 5 shows a formula where the "variable" is the balance in another general ledger account (see Section 7.1.3). The formula in Row 5 evaluates to 0.6 times the balance in ledger account X.

Summarizing, to build any sort of Formula Accounting system one must have a method for representing formulae that allows them to be compared, added, and
evaluated whenever necessary. Ideally, the data structure should allow users to exploit any of the functions found in spreadsheet programs, including @IF in Lotus 123, and to express formulae in terms of combinations of many spreadsheet cells and calls to other programs. However, it is not the intention here to develop a full spreadsheet-like system. All that is needed is a way to represent some reasonably simple formulae in a way that can be readily interpreted by both human beings and computers. The above data structure does that job for this thesis.

8.5 The Prototype Formula Accounting System

*** Figure 8.8 goes approximately here, see next page ***

With the above data structure for representing formulae, the full prototype FA accounting database is represented by the ten Prolog predicates (equivalent to ten relational database tables) shown in Figure 8.8. The two "tables" not yet mentioned are

\[
\text{gl_ac_balance(GL_ACno,Fno)}
\]

\[
\text{price_index(Series, Date, Index_value)}
\]

These are both elementary NIAM fact types, so it is not necessary to draw "dumbbell" diagrams to check against update anomalies. Table gl_ac_balance is used to link chart_of_account records to the many possible "formula" predicates for each account balance. (A "formula" predicate can only have one condition, but many different conditional formulae may be posted to any one ledger account.) Table price_index is used for evaluating formulae for reporting purposes. Straight-
FA Ledger Account Information

chart_of_accounts(GL_ACno, Description, Class).
gl_ac_hierarchy(Parent_class, Child_class, Sequence).
gl_depreciation(Asset_GL_ACno, Accum_depr_ACno, Depr_expense_ACno).
gl_amortization(Asset_GL_ACno, Amort_expense_ACno).
gl_ac_balance(GL_ACno, Fno). /* gl_ac_balance may have many Fnos */

FA Journal Entries

jeh(JEno, Date, Eno, Narration).
jed(JEno, Lno, GL_ACno, Fno). /* jed can have one and only one Fno */

Formulae (for both ledger balances and journal entries)

formula(Fno, LHS_UFno, Condition, RHS_UFno, True_UFno, False_UFno).
uncondit_formula(UFno, Coefficient, Polynomial_term_list).

Price index series (for Inflation Accounting)

price_index(Series, Date, Index_value).

Figure 8.8: Prolog Predicates for the Prototype Formula Accounting system
line interpolation is used for index values between dates in the series. Straight-line extrapolation is used to estimate values on dates outside the range specified.

*** Table 8.2 goes approximately here ***

Table 8.2 describes a number of simple procedures written in Prolog to facilitate use of the prototype Formula Accounting system. The nine-hundred-or-so lines of code for these procedures are reproduced in the first Appendix.

An example of the prototype Formula Accounting system "in action" is presented at the end of the chapter as Exhibit 8.1 (15 pages). The Exhibit is intended to be skimmed, not read with care. What the Exhibit shows is that the combination of the database specified in Figure 8.8 and the procedures specified in Table 8.2 are sufficient to support a workable Formula Accounting system.

*** Figure 8.9 goes approximately here, see next page ***

In case there are any problems reading the formulae in Exhibit 8.1, one last example of actual prototype output may help. Figure 8.9 contains a copy of one journal entry line from the third command in Exhibit 8.1. In the top line of Figure 8.9 is predicate "jed", a journal-entry-detail line. The parameters that follow, still on the top line, show that (i) this is journal entry 1, (ii) line 3, (iii) the

---

8 The prototype system was run on a CDC Cyber 990, which uses a 64-bit word length and computes all figures to 15 significant digits. It would have been possible to round all numbers to four or five significant figures, but the numbers shown are not really a problem.
<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_je(JEno, Date)</td>
<td>Lists all journal entry lines and their formulae for journal entry JEno and evaluates them for date Date. If a formula predicate is unconditional, only its corresponding uncondit_formula predicate is listed.</td>
</tr>
<tr>
<td>list_jes(Date)</td>
<td>Lists all journal entries, evaluating them on date Date.</td>
</tr>
<tr>
<td>list_bal(GL_ACno, Date)</td>
<td>Lists all formulae for the specified account, GL_ACno, and evaluates them for date Date.</td>
</tr>
<tr>
<td>tb(Date)</td>
<td>Lists all account balances. This is effectively a &quot;Trial Balance&quot;. Hence the name, tb.</td>
</tr>
<tr>
<td>money_tb(Date)</td>
<td>Lists account balances as money amounts only, without showing formulae used to calculate them.</td>
</tr>
<tr>
<td>check_balance(JEno)</td>
<td>Checks that the coefficients of all polynomial (and non-polynomial) terms in all journal entry lines for journal entry JEno add to zero (now and forever).</td>
</tr>
<tr>
<td>post_je(JEno)</td>
<td>Posts each journal entry line for journal entry JEno to the appropriate GL_Account by either (i) summing coefficients or inserting new uncondit_formulae for existing formulae, or (ii) inserting new formulae for the GL_Account.</td>
</tr>
<tr>
<td>close_revenue_acs(Date)</td>
<td>On date Date, scans through all revenue and expense GL_Accounts, generating and posting journal entry lines to zero the balance of all revenue and expense GL_Account balance formula, transferring the balance to an account called Profit and Loss. Balance formulae remain in the revenue and expense accounts (but evaluate to zero on Date) unless all coefficients of all variables in the formula are zero.</td>
</tr>
</tbody>
</table>

NB. Uses routines from Interpreter for indexing.

Table 8.2 Simple Procedures for using Formula Accounting
Figure 8.9: A Copy of the Depreciation Expense Journal Entry line from Exhibit 8.1 (p.8-24)
ledger account affected is account number 33407\textsuperscript{11}, and (iv) the Fno for this journal entry line is 108. The \textit{formula} with Fno 108 is listed on the second line of Figure 8.9:

\texttt{formula( 108, 108.<, 109, 110, 111)}

This formula is conditional because depreciation is not allowed to exceed asset cost. The two 108s beside each other have no significance; the first 108 is the Fno, the second is a UFno. The \textit{conditional part} of the formula, underlined above, refers to two unconditional formulae numbered 108 and 109. These and unconditional formulae 110 and 111 are also shown in Figure 8.9. Translated, formula 108 says:

\begin{verbatim}
if d-32953 < 3652.5
then use uncondit_formula 110, i.e., -3221*sp + 0.0978*d*sp
else use uncondit_formula 111, i.e., 357.1*sp.
\end{verbatim}

As with the Lotus example in Chapter 7, days for the variable d are counted from 1 January, 1900, so in the formula above 32953 is equivalent to 23 March, 1990. The right-hand side of the condition, 3652.5, is the number of days in 10 years. Values for the price-index series sp are given on the first page of the Exhibit.

\textsuperscript{11} Following that account number, procedure list\_je has added an account description, in angle brackets. This is for human consumption only.
8.6 Chapter Summary

In this chapter the three data modelling tools to be used in the next chapter were introduced. NIAM diagrams provide a graphic formalism for defining the objects of interest in a database and the facts that relate them. Following the NIAM methodology results in specification of a relational database schema that should be in fifth normal form [Codd, 1970; Kent, 1983]. These relations may then be conveniently stored and manipulated in Prolog.

Using these three steps, i.e., NIAM diagrams to Relational schema to Prolog implementation, a database for Formula Accounting was defined in this chapter in Sections 8.1 through 8.3. For the practical implementation required for testing the REE data model and the Interpreter (to be presented in Chapters 9 and 10, respectively), it was necessary to define a workable Formula Accounting database, and in particular, to decide on a representation for journal-entry and ledger-account-balance formulae. Section 8.4 shows the relatively simple system that was adopted for representing formulae. It should be emphasized that this method of representation was developed for a prototype implementation of Formula Accounting. In practice the representation would need to be more general, and much of the complexity of representation would be hidden from the user. For example, each formula number in the present implementation could appear to the user as a cell in a spreadsheet. Lotus-like "@ functions" could be pre-coded so that complex formulae like that for depreciation in Figure 8.9 would be simple to use.

Finally, Table 8.2 in Section 8.5 defines some useful procedures for displaying and manipulating data in the Formula Accounting database. It must be emphasized
again that the purpose of the prototype is to demonstrate the feasibility of the proposed architecture, not to show how a marketable product would "look and feel". Thus the procedures in Table 8.2 are very simple. No attempt is made to hide the underlying complexity of representing formulae from the user, and no attempt is made to provide the powerful reporting facilities found in most commercial general ledger systems today.

Despite these limitations, a quick skim through Exhibit 8.1 will show that the database as specified in Figure 8.8 and the procedures as specified in Table 8.2 are sufficient to support a workable Formula Accounting system. During the next two chapters it will be helpful to keep in mind that the purpose of the REE database and the Interpreter is to generate FA journal entries for a system at least as powerful as the system described in this chapter.

(4,900 words)
The sequence of commands in the pages that follow is used to display and generate Formula Accounting entries that mirror the journal entries in Figures 7.6 and 7.8 in Chapter 7. The data given in Figure 7.6 were as follows:

Data: On day $d_1$, a firm is established with $c$ cash. Soon after, on day $d_0$, a fixed asset is acquired for $a$. The asset's expected life is $n$ years and its residual value is expected to be zero. Straight-line depreciation is considered appropriate. The end of the first financial year after asset acquisition is day $d_1$, $(d_1-d_0)<=365$. The end of the next financial year is $d_2$. The following are the relevant price index series (all subscripted symbols are constants):

<table>
<thead>
<tr>
<th>day</th>
<th>$d_1$</th>
<th>$d_0$</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>specific prices</td>
<td>$s_1$</td>
<td>$s_0$</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s$</td>
</tr>
<tr>
<td>general prices (CPI)</td>
<td>$g_1$</td>
<td>$g_0$</td>
<td>$g_1$</td>
<td>$g_2$</td>
<td>$s$</td>
</tr>
</tbody>
</table>

As in the Lotus 123 example in Figure 7.2 in the previous chapter, real dates and numbers must be used in place of symbols like "$d_1"", "c"", and "a" above. For this example it is therefore assumed that on 1 March, 1990, proprietor A. MacIntosh established the business with $100,000 cash. Soon after, on 23 March 1990, a 1990 Ford Truck, was acquired for $36,500. The asset's expected life is 10 years and its residual value is expected to be zero. Straight-line depreciation is considered appropriate. Financial year-ends are at 24:00 hours on 31 December 1990, 31 December 1991, etc.. Values for the price index series, $g$ and $sp$, are defined as follows:

```Prosper
price_index(g,900101,100).
price_index(g,940101,120).
price_index(sp,900101,100).
price_index(sp,940101,140).
```

Linear interpolation is to be used for values between those dates where index values are explicitly provided to the system, i.e., between dates 900101 and 940101.

To facilitate comparison with Figures 7.6 and 7.8, the relevant journal entries from those figures have been inserted, in boxes, into the computer output listing below. Commands from the terminal are shown in bold. Comments, numbered 1 to 11, also in bold, indicate what the command is intended to do or show.
1. The command in bold below, \texttt{tb(900101)}), shows that initially, on date 900101, there were no formulae in the ledger; all account balances were zero.

\texttt{tb(900101)}.

Trial Balance for REEFA LTD, date 900101, for Current Cost Accounting, fifo

Overall ledger balance on date 900101 is 0

\texttt{yes}^{10}

2. The command in bold below, \texttt{list\_je(0,910101)}, lists journal entry 0 that records the proprietor's contribution of capital to the business\textsuperscript{11}. The corresponding journal entry from Figure 7.6 is given in the box below. Formulae have been evaluated at arbitrarily selected date 910101. In this case it is simple to check that the coefficients of variables \([I] \) and \([g,1]\) exactly balance, but sometimes it is easier for a human reader to check that a journal entry balances by checking formulae evaluated at some arbitrary date. On 900301 the index value \(g,\_1\) was about 101, so \(c/g,\_1\) (i.e. 100000/991.988) evaluated to 991.988.

\begin{verbatim}
Day d_1:
Dr Bank        	c  
Dr Holding Gain/Loss  	c(g/g,\_1) - 1)  
Cr Capital      	c(g/g,\_1)  

list\_je(0,910101).

jeh(0, 900301, 0,Opening Balances)
jed(0, 1, 11102 <Cash at Bank >, 100).
    uncondit\_formula(100, 100000,[])  

*** On date 910101 formula evaluates to 100000 ***
jed(0, 2, 31401 <Gearing Gain >, 101).
    uncondit\_formula(101, 991.988049972842,[[g, 1]])
    uncondit\_formula(101,-100000,[])

*** On date 910101 formula evaluates to 4155.35035306821 ***
jed(0, 3, 31100 <A. Mackintosh - Capital >, 102).
    uncondit\_formula(102, -991.988049972842,[[g, 1]])

*** On date 910101 formula evaluates to -104155.350353068 ***

yes
\end{verbatim}

\textsuperscript{10} Prolog always says "yes" when a procedure executes successfully. Successfully does not necessarily mean correctly.

\textsuperscript{11} The journal entries in this Appendix were generated by the Interpreter discussed in Chapter 10. However, for the present they should be thought of as having been keyed-in from a keyboard.
3. The command in bold below, listje(1,910101), lists the journal entry to record the purchase of the fixed asset. The corresponding journal entry from Figure 7.6 is shown in the box below (the sequence of the lines is different).

<table>
<thead>
<tr>
<th>Day d₀:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr</td>
<td>Cr Holding Gain/Loss</td>
</tr>
<tr>
<td>Asset</td>
<td>a(s/s₀)</td>
</tr>
<tr>
<td>Holding Gain/Loss</td>
<td>(7.6.4)</td>
</tr>
</tbody>
</table>

listje(1,910101).

jeh(1, 900323, 1, Purchase Equip.)
jed(1, 1, 12801 <1990 Ford Delivery Truck >, 106).
uncondit_formula(106, 357.081157091203,[[sp, 1]])
*** On date 910101 formula evaluates to 39276.4831927144 ***
jed(1, 2, 31501 <CCA Gain above General Prices >, 107).
uncondit_formula(107, 360.997156783105,[[g, 1]])
uncondit_formula(107, -357.081157091203,[[sp, 1]])
*** On date 910101 formula evaluates to -1373.01717592613 ***
jed(1, 3, 33407 <Depreciation Expense - 1990 Ford Truck >, 108).
uncondit_formula(108, 108,<, 109, 110, 111)
uncondit_formula(108, -32953,[[]])
uncondit_formula(109, 3652.5,[[]])
uncondit_formula(110, -3221.60037498326,[[sp, 1]])
uncondit_formula(110, .097763492701219,[[d, 1],[sp, 1]])
uncondit_formula(111, 357.081157091203,[[sp, 1]])
*** On date 910101 formula evaluates to 3053.94147207029 ***
jed(1, 4, 12901 <Accum Depr - 1990 Ford Truck >, 109).
uncondit_formula(109, 112,<, 113, 114, 115)
uncondit_formula(112, -32953,[[]])
uncondit_formula(113, 3652.5,[[]])
uncondit_formula(114, 3221.60037498326,[[sp, 1]])
uncondit_formula(114, .097763492701219,[[d, 1],[sp, 1]])
uncondit_formula(115, -357.081157091203,[[sp, 1]])
*** On date 910101 formula evaluates to -3053.94147207029 ***
jed(1, 5, 11102 <Cash at Bank >, 110).
uncondit_formula(116,-36500,[[]])
*** On date 910101 formula evaluates to -36500 ***
jed(1, 6, 31401 <Gearing Gain >, 111).
uncondit_formula(117, -360.997156783105,[[g, 1]])
uncondit_formula(117, 36500,[[]])
*** On date 910101 formula evaluates to -1403.46601678827 ***

yes
4. The commands in bold below post journal entries 0 and 1 to the ledger then generate a Trial Balance. Command "tb" lists and evaluates account balance formulae. Command "money_tb" evaluates account balance formulae and lists only the money amounts, in cents. Formulae have been evaluated on date 910101.

post_je(0).
yes
post_je(1).
yes
tb(910101).

Trial Balance for REEFA LTD, date 910101, for Current Cost Accounting, fifo

<table>
<thead>
<tr>
<th>GL_AC</th>
<th>Description</th>
<th>Date</th>
<th>Formula Details</th>
</tr>
</thead>
</table>
| 11102  | <Cash at Bank>                  | 910101    | uncondit_formula(103, 63500,[])
|        | *** On date 910101 formula evaluates to 63500 *** Overall account balance on date 910101 is 63500 |
| 12801  | <1990 Ford Delivery Truck>      | 910101    | uncondit_formula(118, 357.081157091203,[sp, 1])
|        | *** On date 910101 formula evaluates to 39276.4831927144 *** Overall account balance on date 910101 is 39276.4831927144 |
| 12901  | <Accum Depr - 1990 Ford Truck>  | 910101    | uncondit_formula(122.6, 3221.60037498326,[sp, 1])
|        | *** On date 910101 formula evaluates to -3053.94147207029 *** Overall account balance on date 910101 is -3053.94147207029 |
| 31100  | <A. Mackintosh - Capital>       | 910101    | uncondit_formula(105, -991.988049972842,[])
|        | *** On date 910101 formula evaluates to -104155.350353068 *** Overall account balance on date 910101 is -104155.350353068 |
| 31401  | <Gearing Gain>                  | 910101    | uncondit_formula(104, 630.990893189737,[])
|        | *** On date 910101 formula evaluates to 2751.88433627971 *** Overall account balance on date 910101 is 2751.88433627971 |
| 31501  | <CCA Gain above General Prices> | 910101    | uncondit_formula(119, 360.997156783105,[])
|        | *** On date 910101 formula evaluates to -1373.01717592613 *** Overall account balance on date 910101 is -1373.01717592613 |
| 33407  | <Depreciation Expense - 1990 Ford Truck> | 910101 | uncondit_formula(120, -3221.60037498326,[])
|        | *** On date 910101 formula evaluates to -1373.01717592613 *** Overall account balance on date 910101 is -1373.01717592613 |
**uncondit_formula(123, 357.081157091203,[[sp, 1]])**

*** On date 910101 formula evaluates to 3053.94147207029 ***

Overall account balance on date 910101 is 3053.94147207029

Overall ledger balance on date 910101 is -4.6566128730774E-0010

---

money_{tb}(910101).

**MONEY TRIAL BALANCE, REEFA LTD, as at 910101**

for Current Cost Accounting, fifo

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance (cents)</th>
<th>CR Balance (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11102</td>
<td>Cash at Bank</td>
<td>6350000</td>
<td></td>
</tr>
<tr>
<td>12801</td>
<td>1990 Ford Delivery Truck</td>
<td>3927648</td>
<td>-305394</td>
</tr>
<tr>
<td>12901</td>
<td>Accum Depr - 1990 Ford Truck</td>
<td></td>
<td>-10415535</td>
</tr>
<tr>
<td>31100</td>
<td>A. Mackintosh - Capital</td>
<td></td>
<td>-137302</td>
</tr>
<tr>
<td>31401</td>
<td>Gearing Gain</td>
<td>275188</td>
<td></td>
</tr>
<tr>
<td>31501</td>
<td>CCA Gain above General Prices</td>
<td></td>
<td>-10858231</td>
</tr>
<tr>
<td>33407</td>
<td>Depreciation Expense - 1990 Ford Truck</td>
<td>305394</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10858230</td>
</tr>
</tbody>
</table>

*************** ERROR ****************************

TRIAL BALANCE DOES NOT BALANCE.

Difference is -1 cents.

*************** ERROR ****************************

yes
The commands in bold below first generate and post, then list, the closing journal entry for 910101. Three revenue and expense accounts are affected: Gearing Gain, CCA Gain above General Prices, and Depreciation Expense. The entry from Figure 7.6 corresponding to jed lines 5, 6, and 7 is shown in the box. Note that line 6 is an unconditional formula, so that after posting there are two "formula" predicates associated with account 33305, one conditional and one unconditional. These are shown in the last few lines of the tb listing on the next page (*** lines $3053.94 and -$3053.94). (p.8-29)

| Day d₁: |
|---|---|
| Dr Profit and Loss | B₁*(g/g₁) | (7.6.10) |
| Dr Holding Gain/Loss | B₁*(s/s₁ - g/g₁) | (7.6.11) |
| Cr Depreciation Expense | B₁*(s/s₁) | (7.6.12) |

(Where B₁ is a(s₁/s₀)/(365*n)*(d₁-d₀), a constant, being the current balance in the Depreciation Expense account on day d₁.)

```
close_revenue_acs(910101).
Entering close_revenue_acs for period ending 910101
yes

list_je(2,910101).

jeh( 2, 910101, closing_acs, Closing revenue and expense for period)
  jed( 2, 1, 31900 <Profit and Loss >, 116).
  uncondit_formula(128, 26.2092765013342,[[g, 1]])
  *** On date 910101 formula evaluates to 2751.88433627969 ***
  jed( 2, 2, 31401 <Gearing Gain >, 117).
  uncondit_formula(129, -26.2092765013342,[[g, 1]])
  *** On date 910101 formula evaluates to -2751.88433627969 ***
  jed( 2, 3, 31900 <Profit and Loss >, 118).
  uncondit_formula(130, -13.0767802739771,[[g, 1]])
  *** On date 910101 formula evaluates to -1373.01717592612 ***
  jed( 2, 4, 31501 <CCA Gain above General Prices >, 119).
  uncondit_formula(131, 13.0767802739771,[[g, 1]])
  *** On date 910101 formula evaluates to 1373.01717592612 ***
  jed( 2, 5, 31900 <Profit and Loss >, 120).
  uncondit_formula(132, 29.0861048937074,[[g, 1]])
  *** On date 910101 formula evaluates to 3053.94147207028 ***
  jed( 2, 6, 33407 <Depreciation Expense - 1990 Ford Truck >, 121).
  uncondit_formula(133, -27.7648319271607,[[sp, 1]])
  *** On date 910101 formula evaluates to -3053.94147207028 ***
  jed( 2, 7, 31501 <CCA Gain above General Prices >, 122).
  uncondit_formula(134, -29.0861048937074,[[g, 1]])
  uncondit_formula(134, 27.7648319271607,[[sp, 1]])
  *** On date 910101 formula evaluates to 0 ***
```

Yes
Trial Balance after closing revenue and expense accounts. Note that compared to the report two pages earlier, a Profit and Loss account has appeared and that the balances for Gearing Gain, CCA Gain above General Prices, and Depreciation Expense are effectively zero (highlighted in bold) on 910101.

\( \text{tb}(910101) \).

Trial Balance for REEFA LTD, date 910101, for Current Cost Accounting, fifo

\[ \begin{align*}
\text{GL}_\text{AC} = 11102 <\text{Cash at Bank} >, \\
\text{uncondit}_\text{formula}(103, 63500,[[]) \\
\text{*** On date } 910101 \text{ formula evaluates to } 63500 \text{ ***} \\
\text{Overall account balance on date } 910101 \text{ is } 63500 \\
\text{GL}_\text{AC} = 12801 <1990 \text{ Ford Delivery Truck} >, \\
\text{uncondit}_\text{formula}(118, 357.081157091203,[[sp, 1]]) \\
\text{*** On date } 910101 \text{ formula evaluates to } 39276.4831927144 \text{ ***} \\
\text{Overall account balance on date } 910101 \text{ is } 39276.4831927144 \\
\text{GL}_\text{AC} = 12901 <\text{Accum Depr - 1990 Ford Truck} >, \\
\text{formula}(115, 125,<, 126, 124, 127) \\
\text{uncondit}_\text{formula}(125, -32953,[[]) \\
\text{uncondit}_\text{formula}(125, 1,[[d, 1]]) \\
\text{uncondit}_\text{formula}(126, 3652.5,[[]) \\
\text{uncondit}_\text{formula}(124, 3221.60037498326,[[sp, 1]]) \\
\text{uncondit}_\text{formula}(124, -.097763492701219,[[d, 1],[sp, 1]]) \\
\text{uncondit}_\text{formula}(127, -357.081157091203,[[sp, 1]]) \\
\text{*** On date } 910101 \text{ formula evaluates to } -3053.94147207029 \text{ ***} \\
\text{Overall account balance on date } 910101 \text{ is } -3053.94147207029 \\
\text{GL}_\text{AC} = 31100 <\text{A. Mackintosh - Capital} >, \\
\text{uncondit}_\text{formula}(105, -991.988049972842,[[g, 1]]) \\
\text{*** On date } 910101 \text{ formula evaluates to } -104155.350353068 \text{ ***} \\
\text{Overall account balance on date } 910101 \text{ is } -104155.350353068 \\
\text{GL}_\text{AC} = 31401 <\text{Gearing Gain} >, \\
\text{uncondit}_\text{formula}(104, -63500,[[]) \\
\text{uncondit}_\text{formula}(104, 604.781616688404,[[g, 1]]) \\
\text{*** On date } 910101 \text{ formula evaluates to } 2.3283064365387E-0010 \text{ ***} \\
\text{Overall account balance on date } 910101 \text{ is } 2.3283064365387E-0010 \\
\text{GL}_\text{AC} = 31501 <\text{CCA Gain above General Prices} >, \\
\text{uncondit}_\text{formula}(119, 344.987832163375,[[g, 1]]) \\
\text{uncondit}_\text{formula}(119, -329.316325164044,[[sp, 1]]) \\
\text{*** On date } 910101 \text{ formula evaluates to } 0 \text{ ***} \\
\text{Overall account balance on date } 910101 \text{ is } 0 \\
\text{GL}_\text{AC} = 31900 <\text{Profit and Loss} >, \\
\text{uncondit}_\text{formula}(135, 42.2186011210645,[[g, 1]]) \\
\text{*** On date } 910101 \text{ formula evaluates to } 4432.80863242384 \text{ ***} \\
\text{Overall account balance on date } 910101 \text{ is } 4432.80863242384 \\
\text{GL}_\text{AC} = 33407 <\text{Depreciation Expense - 1990 Ford Truck} >, \\
\text{formula}(114, 121,<, 122, 120, 123) \\
\text{uncondit}_\text{formula}(121, -32953,[[]) \\
\text{uncondit}_\text{formula}(121, 1,[[d, 1]]) \\
\text{uncondit}_\text{formula}(122, 3652.5,[[]) \\
\text{uncondit}_\text{formula}(120, -3221.60037498326,[[sp, 1]]) \\
\text{uncondit}_\text{formula}(120, .097763492701219,[[d, 1],[sp, 1]]) \\
\end{align*} \]
MONEY TRIAL BALANCE, REEFA LTD, as at 910101
for Current Cost Accounting, fifo

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance</th>
<th>CR Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>11102</td>
<td>Cash at Bank</td>
<td>6350000</td>
<td></td>
</tr>
<tr>
<td>12801</td>
<td>1990 Ford Delivery Truck</td>
<td>3927648</td>
<td></td>
</tr>
<tr>
<td>12901</td>
<td>Accum Depr - 1990 Ford Truck</td>
<td></td>
<td>-305394</td>
</tr>
<tr>
<td>31100</td>
<td>A. Mackintosh - Capital</td>
<td></td>
<td>-10415535</td>
</tr>
<tr>
<td>31900</td>
<td>Profit and Loss</td>
<td>443281</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10720929</td>
<td>-10720929</td>
</tr>
</tbody>
</table>

Overall ledger balance on date 910101 is -4.6566128730774E-0010

Overall account balance on date 910101 is 1.4551915228367E-0011
7. The formulae in this trial balance are unchanged from the previous page, but the evaluation date is one year later, so the balance in Gearing Gain, CCA Gain above General Prices, and Depreciation Expense are non-zero.

tb(920101).

Trial Balance for REEFA LTD, date 920101, for Current Cost Accounting, fifo

GL_AC = 11102 <Cash at Bank >,
  uncondit_formula(103, 63500,[])
  *** On date 920101 formula evaluates to 63500 ***
  Overall account balance on date 920101 is 63500

GL_AC = 12801 <1990 Ford Delivery Truck >,
  uncondit_formula(118, 357.081157091203,[[sp, 1]])
  *** On date 920101 formula evaluates to 42844.8506763089 ***
  Overall account balance on date 920101 is 42844.8506763089

GL_AC = 12901 <Accum Depr - 1990 Ford Truck >,
  formula(115, 125,<, 126, 124, 127)
  uncondit_formula(125,-32953,[])
  uncondit_formula(125, 1,[[d, 1]])
  uncondit_formula(126, 3652.5,[])
  uncondit_formula(124, .097763492701219,[[d, 1],[[sp, 1]]])
  uncondit_formula(127, -357.081157091203,[[sp, 1]])
  *** On date 920101 formula evaluates to -7624.68252966367 ***
  Overall account balance on date 920101 is -7624.68252966367

GL_AC = 31100 <A. Mackintosh - Capital >,
  uncondit_formula(105, -991.988049972842,[[g, 1]])
  *** On date 920101 formula evaluates to -109111.895708853 ***
  Overall account balance on date 920101 is -109111.895708853

GL_AC = 31401 <Gearing Gain >,
  uncondit_formula(104,-63500,[])
  *** On date 920101 formula evaluates to 3021.8383311606 ***
  Overall account balance on date 920101 is 3021.8383311606

GL_AC = 31501 <CCA Gain above General Prices >,
  uncondit_formula(119, 344.987832163375,[[g, 1]])
  uncondit_formula(119, -329.316325164044,[[sp, 1]])
  *** On date 920101 formula evaluates to -1567.15069993329 ***
  Overall account balance on date 920101 is -1567.15069993329

GL_AC = 31900 <Profit and Loss >,
  uncondit_formula(135, 42.2186011210645,[[g, 1]])
  *** On date 920101 formula evaluates to 4643.75715274125 ***
  Overall account balance on date 920101 is 4643.75715274125

GL_AC = 33407 <Depreciation Expense - 1990 Ford Truck >,
  formula(114, 121,<, 122, 120, 123)
  uncondit_formula(121,-32953,[])
  uncondit_formula(121, 1,[[d, 1]])
  uncondit_formula(122, 3652.5,[])
  uncondit_formula(120, -3221.60037498326,[[sp, 1]])
  uncondit_formula(120, .097763492701219,[[d, 1],[[sp, 1]]])
  uncondit_formula(123, 357.081157091203,[[sp, 1]])
  *** On date 920101 formula evaluates to 7624.68252966367 ***
**uncondit_formula(136, -27.7648319271607,[[sp, 1]])**

*** On date 920101 formula evaluates to -3331.39975142453 ***

Overall account balance on date 920101 is 4293.28277823914

Overall ledger balance on date 920101 is -4.6566128730774E-0010

money_tb(920101).

MONEY TRIAL BALANCE, REEFA LTD, as at 920101
for Current Cost Accounting, fifo

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance (cents)</th>
<th>CR Balance (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11102</td>
<td>Cash at Bank</td>
<td>6350000</td>
<td></td>
</tr>
<tr>
<td>12801</td>
<td>1990 Ford Delivery Truck</td>
<td>4284485</td>
<td></td>
</tr>
<tr>
<td>12901</td>
<td>Accum Depr - 1990 Ford Truck</td>
<td></td>
<td>-762468</td>
</tr>
<tr>
<td>31100</td>
<td>A. Mackintosh - Capital</td>
<td></td>
<td>-10911190</td>
</tr>
<tr>
<td>31401</td>
<td>Gearing Gain</td>
<td>302184</td>
<td></td>
</tr>
<tr>
<td>31501</td>
<td>CCA Gain above General Prices</td>
<td></td>
<td>-156715</td>
</tr>
<tr>
<td>31900</td>
<td>Profit and Loss</td>
<td>464376</td>
<td></td>
</tr>
<tr>
<td>33407</td>
<td>Depreciation Expense - 1990 Ford Truck</td>
<td>429328</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11830373</td>
</tr>
</tbody>
</table>

```yes```
8. The commands in bold below first generate and post, then list, the closing journal entry for 920101. The journal entry from Figure 7.6 corresponding to lines 5 through 7 (the conditional part of account 33305's balance formula) and lines 8 through 10 (the unconditional part) is shown in the box.

| Day d₂: |
|------------------|------------------|------------------|
| Dr Profit and Loss | B₂*(g/g₂) | (7.6.13) |
| Dr Holding Gain/Loss | B₂*(s/s₂ - g/g₂) | (7.6.14) |
| Cr Depreciation Expense | B₂*(s/s₂) | (7.6.15) |

(Where B₂ is a(s₂/s₀)/(365*n)*(d₂-d₁), a constant, being the current balance in the Depreciation Expense account on day d₂.)

```
close_revenue_acs(920101).
Entering close_revenue_acs for period ending 920101
yes
```
```
list_j(3,920101).
```
```
jeh(3, 1, 31900 <Profit and Loss >, 125).
uncondit_formula(137, 27.4729670306514,[[g, 1]])
*** On date 920101 formula evaluates to 3021.83833116059 ***
jed(3, 2, 31401 <Gearing Gain >, 126).
uncondit_formula(138, -27.4729670306514,[[g, 1]])
*** On date 920101 formula evaluates to -3021.83833116059 ***
jed(3, 3, 31900 <Profit and Loss >, 127).
uncondit_formula(139, -14.2477110927352,[[g, 1]])
*** On date 920101 formula evaluates to -1567.15069993329 ***
jed(3, 4, 31501 <CCA Gain above General Prices >, 128).
uncondit_formula(140, 69.3196090593569,[[g, 1]])
*** On date 920101 formula evaluates to 7624.68252966361 ***
jed(3, 5, 31900 <Profit and Loss >, 129).
uncondit_formula(141, 63.5462702557825,[[sp, 1]])
*** On date 920101 formula evaluates to 2.9103830456734E-0011 ***
jed(3, 6, 33407 <Depreciation Expense - 1990 Ford Truck >, 130).
uncondit_formula(142, -30.2873368813396,[[g, 1]])
*** On date 920101 formula evaluates to -3331.39975142451 ***
jed(3, 7, 31501 <CCA Gain above General Prices >, 131).
uncondit_formula(143, -27.7648319271606,[[g, 1]])
uncondit_formula(144, 27.7648319271606,[[g, 1]])
*** On date 920101 formula evaluates to 0 ***
jed(3, 8, 31900 <Profit and Loss >, 132).
uncondit_formula(145, -30.2873368813396,[[g, 1]])
*** On date 920101 formula evaluates to -3331.39975142451 ***
jed(3, 9, 33407 <Depreciation Expense - 1990 Ford Truck >, 133).
uncondit_formula(146, -27.7648319271606,[[g, 1]])
uncondit_formula(147, -27.7648319271606,[[sp, 1]])
*** On date 920101 formula evaluates to 0 ***
```
```
yes
```
9. Trial Balance after closing revenue and expense accounts on 920101. All revenue and expense accounts evaluate to effectively zero (highlighted in bold) on 920101.

tb(920101).

Trial Balance for REEFA LTD, date 920101, for Current Cost Accounting, fifo

GL_AC = 11102 <Cash at Bank >,
  uncondit_formula(103, 63500,[])
  *** On date 920101 formula evaluates to 63500 ***
  Overall account balance on date 920101 is 63500
GL_AC = 12801 <1990 Ford Delivery Truck >,
  uncondit_formula(118, 357.081157091203,[])
  *** On date 920101 formula evaluates to 42844.8506763089 ***
  Overall account balance on date 920101 is 42844.8506763089
GL_AC = 12901 <Accum Depr - 1990 Ford Truck >,
  formula(115, 125,<, 126, 124, 127)
  uncondit_formula(125,-32953,[])
  uncondit_formula(126, 3652.5,[])
  uncondit_formula(124, 3221.60037498326,[])
  *** On date 920101 formula evaluates to -7624.6825966367 ***
  Overall account balance on date 920101 is -7624.6825966367
GL_AC = 31100 <A. Mackintosh - Capital >,
  uncondit_formula(105,-991.988049972842,[])
  *** On date 920101 formula evaluates to -109111.895708853 ***
  Overall account balance on date 920101 is -109111.895708853
GL_AC = 31401 <Gearing Gain >,
  uncondit_formula(104,-63500,[])
  *** On date 920101 formula evaluates to 2.3283064365387E-0010 ***
  Overall account balance on date 920101 is 2.3283064365387E-0010
GL_AC = 31501 <CCA Gain above General Prices >,
  uncondit_formula(119, 320.203271078091,[])
  *** On date 920101 formula evaluates to -2.3283064365387E-0010 ***
  Overall account balance on date 920101 is -2.3283064365387E-0010
GL_AC = 31900 <Profit and Loss >,
  uncondit_formula(135, 94.4761292369981,[])
  *** On date 920101 formula evaluates to 10391.7275622077 ***
  Overall account balance on date 920101 is 10391.7275622077
GL_AC = 33407 <Depreciation Expense - 1990 Ford Truck >,
  formula(114, 121,<, 122, 120, 123)
  uncondit_formula(121,-32953,[])
  uncondit_formula(122, 3652.5,[])
  uncondit_formula(120, 3221.60037498326,[])
  *** On date 920101 formula evaluates to 7624.6825966367 ***
uncondit_formula(136, -63.5462702557829,[[sp, 1]])

*** On date 920101 formula evaluates to -7624.6825296637 ***

Overall account balance on date 920101 is -2.9103830456734E-0011

Overall ledger balance on date 920101 is -4.6566128730774E-0010

money_tb(920101).

MONEY TRIAL BALANCE, REEFA LTD, as at 920101
for Current Cost Accounting, fifo

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance (cents)</th>
<th>CR Balance (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11102</td>
<td>Cash at Bank</td>
<td>6350000</td>
<td></td>
</tr>
<tr>
<td>12801</td>
<td>1990 Ford Delivery Truck</td>
<td>4284485</td>
<td></td>
</tr>
<tr>
<td>12901</td>
<td>Accum Depr - 1990 Ford Truck</td>
<td></td>
<td>-762468</td>
</tr>
<tr>
<td>31100</td>
<td>A. Mackintosh - Capital</td>
<td></td>
<td>-10911190</td>
</tr>
<tr>
<td>31900</td>
<td>Profit and Loss</td>
<td>1039173</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11673658</td>
<td>-11673658</td>
</tr>
</tbody>
</table>

8 - 34
The command in bold below, list_je(4,930101), shows the journal entry for the sale of the fixed asset on 16 August, 1992, for $15,000 cash. The journal entry from Figure 7.8 is shown in the box, with annotation in bold indicating the jeds used in the prototype system (no Sale of Asset account was used).

### Six FAJE{s} for day \(d_3\):

<table>
<thead>
<tr>
<th>Description</th>
<th>Equation</th>
<th>JED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr Profit and Loss</td>
<td>(B_3 \times (g/g_3))</td>
<td>jed(4,3,...) (7.8.1)</td>
</tr>
<tr>
<td>Cr Holding gain/loss</td>
<td>(B_3 \times (s/s_3 - g/g_3))</td>
<td>jed(4,4,...) (7.8.2)</td>
</tr>
<tr>
<td>Cr Depreciation Expense</td>
<td>(B_3 \times (s/s_2))</td>
<td></td>
</tr>
<tr>
<td>(Where (B_3 = \frac{a(s_3/s_0)}{(365\times n)} \times (d_3 - d_2)), a constant.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr Sales of Asset</td>
<td>(a(s/s_0))</td>
<td>jed(4,7,...) (7.8.4)</td>
</tr>
<tr>
<td>Cr Asset</td>
<td>(a(s/s_0))</td>
<td></td>
</tr>
<tr>
<td>Dr Accumulated Depreciation</td>
<td>(a(s/s_0)/(365\times n) \times (d-d_0))</td>
<td>jed(4,8,...) (7.8.6)</td>
</tr>
<tr>
<td>Cr Sale of Asset</td>
<td>(a(s/s_0)/(365\times n) \times (d-d_0))</td>
<td></td>
</tr>
<tr>
<td>Dr Sales of Asset</td>
<td>(a(s/s_0)/(365\times n) \times (d-d_3))</td>
<td>jed(4,5,...) (7.8.8)</td>
</tr>
<tr>
<td>Cr Depreciation Expense</td>
<td>(a(s/s_0)/(365\times n) \times (d-d_3))</td>
<td>jed(4,6,...) (7.8.9)</td>
</tr>
<tr>
<td>Dr Holding gain/loss</td>
<td>(B_4) (see Figure 7.8)</td>
<td></td>
</tr>
<tr>
<td>Cr Sale of Asset</td>
<td>(B_4) (see Figure 7.8)</td>
<td></td>
</tr>
<tr>
<td>Dr Cash</td>
<td>(b)</td>
<td>jed(4,1,...) (7.8.12)</td>
</tr>
<tr>
<td>Dr Holding gain/loss</td>
<td>(b(g/g_3 - 1))</td>
<td>jed(4,2,...) (7.8.13)</td>
</tr>
<tr>
<td>Cr Sale of Asset</td>
<td>(b(g/g_3))</td>
<td></td>
</tr>
</tbody>
</table>

After posting the above FAJE{s}, the final (credit) balance of the Sale of Asset account is a linear function of \(g\): 

\[-a(g/g_0) + b(g/g_0) + a/(365\times n) \times [(s_1/s_0)(d_1-d_0)/(g/g_3) + (s_2/s_0)(d_2-d_1)/(g/g_2) + (s_3/s_0)(d_3-d_2)/(g/g_3)]\]

---

list_je(4,930101).
jed( 4, 5, 33407 <Depreciation Expense - 1990 Ford Truck >, 139).

uncondit_formula(139, 151,<, 152, 153, 154)

uncondit_formula(151, 1,[d, 1])

uncondit_formula(152, 3652.5,[l])

uncondit_formula(153, 3221.6003492701219,[d, 1],[sp, 1])

uncondit_formula(154, -357.081157091203,[sp, 1])

*** On date 930101 formula evaluates to -12897.8552860525 ***

jed( 4, 6, 33407 <Depreciation Expense - 1990 Ford Truck >, 140).

uncondit_formula(155, 63.462702557829,[sp, 1])

*** On date 930101 formula evaluates to 8259.71028170642 ***

jed( 4, 7, 12801 <1990 Ford Delivery Truck >, 141).

uncondit_formula(156, -357.081157091203,[sp, 1])

*** On date 930101 formula evaluates to -46413.2181599033 ***

jed( 4, 8, 12901 <Accum Depr - 1990 Ford Truck >, 142).

formular( 142, 157,<, 158, 159, 160)

uncondit_formula(157, -32953,[l])

uncondit_formula(158, 3652.5,[l])

uncondit_formula(159, 0,[d, 1],[sp, 1])

uncondit_formula(160, 0,[sp, 1])

*** On date 930101 formula evaluates to 12897.8552860525 ***

jed( 4, 9, 32500 <Non-operating Income >, 143).

uncondit_formula(161, 0,[l])

uncondit_formula(161, -157.388154454268,[g, 1])

uncondit_formula(161, 293.534886835421,[sp, 1])

*** On date 930101 formula evaluates to 20055.4860107661 ***

jed( 4, 10, 32500 <Non-operating Income >, 144).

formular( 144, 162,<, 163, 164, 165)

uncondit_formula(162, 1,[d, 1])

uncondit_formula(162, -32953,[l])

uncondit_formula(163, 3652.5,[l])

uncondit_formula(164, 0,[d, 1],[sp, 1])

uncondit_formula(164, 0,[sp, 1])

uncondit_formula(165, 0,[sp, 1])

*** On date 930101 formula evaluates to 0 ***

yes
11. Trial balance after the previous (sale of asset) journal entry has been posted to the ledger. Note that the Asset, Accumulated Depreciation, and Depreciation Expense accounts have disappeared (as they should have).

tb(930101).

Trial Balance for REEFA LTD, date 930101, for Current Cost Accounting, fifo

GL_AC = 11102 <Cash at Bank >,
    uncondit_formula(103, 78500,[])
*** On date 930101 formula evaluates to 78500 ***
Overall account balance on date 930101 is 78500

GL_AC = 31100 <A Mackintosh - Capital >,
    uncondit_formula(105, -991.988049972842,[[g, 1]])
*** On date 930101 formula evaluates to -114068.441064638 ***
Overall account balance on date 930101 is -114068.441064638

GL_AC = 31401 <Gearing Gain >,
    uncondit_formula(104, 709.933921988908,[[g, 1]])
    uncondit_formula(104,-78500,[])
*** On date 930101 formula evaluates to 3135.11217942182 ***
Overall account balance on date 930101 is 3135.11217942182

GL_AC = 31501 <CCA Gain above General Prices >,
    uncondit_formula(119, 320.203271078091,[[g, 1]])
    uncondit_formula(119,-293.534886835421,[[sp, 1]])
*** On date 930101 formula evaluates to -1333.41921213362 ***
Overall account balance on date 930101 is -1333.41921213362

GL_AC = 31900 <Profit and Loss >,
    uncondit_formula(135, 119.239011360112,[[g, 1]])
*** On date 930101 formula evaluates to 13711.2620865837 ***
Overall account balance on date 930101 is 13711.2620865837

GL_AC = 32500 <Non-operating Income >,
    uncondit_formula(166, 0,[])
    uncondit_formula(166, -157.388154454268,[[g, 1]])
    uncondit_formula(166, 293.534886835421,[[sp, 1]])
*** On date 930101 formula evaluates to 20055.4860107661 ***
formula(146, 168,<, 169, 167, 170)
    uncondit_formula(168, 1,[[d, 1]])
    uncondit_formula(168,-32953,[])
    uncondit_formula(169, 3652.5,[])
    uncondit_formula(167, 0,[[d, 1],[sp, 1]])
    uncondit_formula(167, 0,[[sp, 1]])
    uncondit_formula(170, 0,[[sp, 1]])
*** On date 930101 formula evaluates to 0 ***
Overall account balance on date 930101 is 20055.4860107661

Overall ledger balance on date 930101 is -4.65661287307774E-010
=================================================================
yes

Note: The "action" parts (i.e., the do-if-true, and do-if-false parts) of Formula 146, ten lines from the bottom, are unconditional formulae 167 and 170. Both have coefficients of zero, so formula 146 must always evaluate to zero, and this formula should have been deleted from the account. It did not seem worth making refinements like this for the prototype system.
In the proposed computer-based accounting system architecture, reproduced in Figure 9.1, the REE data model and the Interpreter provide a link between a firm's special-purpose transaction processing systems, on the left, and the Formula Accounting general ledger system, on the right. It is assumed that future accounting systems will capture most data of economic significance in the firm's special-purpose transaction processing systems. It is also assumed that these subsystems are likely to differ so much from organization to organization that few useful generalizations can be made about them. Consequently, it was decided that this thesis should focus on the process of converting details of economic events captured by those heterogeneous subsystems into accounting reports for income determination and "wealth" measurement. This task is common to many organizations.

Figure 9.1: The Proposed Architecture for Computer-based Accounting
The architecture depicted in Figure 9.1 relies on Formula Accounting to enable it to deal with the problem of asset and liability values that change, directly or indirectly, with time. This problem, which becomes most pressing in times of high inflation, was identified in Chapter 6 as one of the major difficulties in accounting practice today. As demonstrated in the last two chapters, Formula Accounting uses the computer's ability to quickly evaluate formulae at any given report date to provide a reasonably straight-forward, reasonably inflation-tolerant, system of accounting.

It is now time to consider linking the special-purpose subsystems on the left of Figure 9.1 and the Formula Accounting general ledger system on the right. Demand for such links is likely to be high. For instance, it was shown in Chapter 4 that in a large proportion of present-day accounting systems data are fed into the general ledger directly from subsystems like those on the left of Figure 9.1. Thus the question addressed in this chapter and the next is as follows:

How can the data from the heterogeneous subsystems depicted on the left of Figure 9.1 be used to generate Formula Accounting journal entries?

In essence, there are only two ways to answer this question. One is for the subsystems themselves to generate Formula Accounting journal entries. The other is to construct some sort of front-end software that draws upon data in the subsystem databases as it generates FA journal entries. The former approach, depicted in the upper path in Figure 9.1, might well be the better
option in practice because the subsystems have ready access to their own databases, and in many cases the formulae they must generate are relatively simple. The latter approach, depicted in the lower path in Figure 9.1, has the advantage that it centralizes all the income and wealth measurement rules in the program called the Interpreter. However, access to the data is likely to be more difficult than it would be for the subsystem programs themselves. Now, since the data structures of the subsystem databases are unknown, the only feasible approach to investigate in this thesis is the second one, i.e., the lower path in Figure 9.1. As discussed in Chapter 6, this is not a restriction because the rules for generating formulae are equally applicable whether they are encoded in one central Interpreter or distributed over the various subsystems.

Specification of the data requirements for the Interpreter is no simple matter. The concepts modelled need to be sufficiently general to be applicable in any situation where accounting income and wealth are measured. Investigation of these concepts probes the very core of the accounting measurement process. That investigation indicates that accounting income and wealth measurement depend on knowledge of two fundamental sorts of information: knowledge about resources owned or controlled by a firm, and knowledge about exchange events whereby the firm changes the set of resources under its control. This led to the hypothesis that information about resources and exchange events is necessary and sufficient for the generation of accounting income and wealth measures.

In a sense, the remainder of this thesis tests that hypothesis. This chapter defines what resources and exchange events are, and what information about them must be captured for accounting reports to be generated. The definition
of what data should be captured is expressed in the form of a data model called the Resources and Exchange Events (REE) model. The next chapter then outlines the rules used by accountants when converting facts from an REE data model into Formula Accounting journal entries for income and wealth measurement. These rules are, of course, encoded in the Interpreter program shown to the right of the REE model in Figure 9.1.

9.1 The Resources and Exchange Events Model: An Overview

In terms of prior work, REE may be viewed as a logical extension of Ijiri's work on Multi-dimensional Accounting [1967, 1975]. Much of the initial structure for REE was developed independently of Ijiri's work, but as Ijiri had developed a more coherent proposal for Multi-dimensional Accounting fifteen years before this study commenced, the credit for the basic framework of Asset and Obligation accounts and Exchange Events must go to him. REE also has many similarities to McCarthy's REA accounting model [1982] (which also draws upon Ijiri's work) and follows very careful consideration of McCarthy's work [1979, 1982]. REE and McCarthy's model, REA, are compared in detail in Section 9.4.

In terms of Figure 9.1, the REE model is intended to be definable as a view of the facts recorded in the various subsystem databases. It is also intended to be a minimal description of resources and exchange events: to qualify for inclusion in the REE model, data must be necessary for the Interpreter to generate its Formula Accounting journal entries.
The REE model consists of two parts, an Exchange Events part, which contains a
detailed historical record of exchange events recorded in the format specified
below, and a Resources part. Resources are objects that an entity would like to
place under its control. They have utility and they are scarce [Ijiri, 1975,
p.52].

The Exchange Events part of the REE model, described more fully in Section 9.2
below, is concerned with two main object types. These are (i) Asset and
Obligation accounts, and (ii) Exchange Events themselves. Asset and Obligation
accounts (AO Accounts) record the number of units of a resource controlled by
an entity or promised to some other entity. Each AO Account describes one,
and only one, resource type. Exchange Events record changes in the quantities
of resources recorded in different Asset and Obligation accounts. To an
accountant, the Exchange Events part of the REE model looks just like a
conventional bookkeeping system with ledger accounts and journal entries. The
only difference is that AO Accounts are only "balance sheet" type accounts
(there are no revenue and expense accounts), and each AO Account is kept in
its own unit of measure, e.g., units, kilos, litres, dollars, or Yen, corresponding
to the resource being described.

1 A full set of definitions of terms used in this chapter, mainly from Ijiri [1975]
and SFAC 6 [FASB, 1986], is included in Appendix 9.1. Most of these definitions have
quite well-accepted meanings in the literature, so they have not been included in the
body of the thesis.

2 It is awkward to have to refer to "asset and obligation" accounts, "balance
sheet" accounts, and "revenue and expense" accounts, but there are no single-word
equivalents. Sprague [1907, p.68] calls "balance sheet" accounts the "Specific"
accounts, and "revenue and expense" accounts the "Economic" accounts, but these
terms are not used today.
Since a typical Exchange Event will "debit" one AO Account in one unit of measure, and "credit" another AO Account in another unit of measure, there is no concept of a balanced journal entry in REE. However, REE Exchange Events are used to generate Formula Accounting journal entries. They must therefore record sufficient information about the real world events so that once some valuation rules have been agreed upon, it is possible to deduce balanced double-entry Formula Accounting journal entries to describe those events. For instance, if an item of inventory, X, is purchased for $10 cash, under historical cost valuation rules it is possible to deduce that the value to be placed on the X is $10, and the journal entry follows automatically. Rules for deducing values for journal entries are discussed in the next chapter, Chapter 10.

The second part of the REE model describes Resources. Here, the resources that are accounted for in AO Accounts are described. Resources prove rather more difficult to define than many theorists have realized because they have a perverse habit of changing both their nature and value over time. For instance, a bunch of flowers in a florist's inventory will wilt in a week or so. Is a wilted bunch of flowers the same resource that was purchased a week ago? Physically many of the atoms are the same. But in the florist's mind it is no longer saleable inventory, and in that sense the resource has changed. Equally difficult, if item X is sold to a customer on credit for $20, how is that customer's promise to pay $20 to be compared to $20 in the bank? Are the two resources equivalent, or do the different levels of risk make them different resources? Figure 9.3 (in Section 9.2) and Section 9.3 present and discuss the
Resource classification scheme used in the prototype REE system described in Chapter 10. A similar approach could be used in any REE model.

In the remainder of the chapter, Section 9.2 describes the Exchange Events part of REE, and Section 9.3 describes the Resources part. Section 9.4 then compares the REE model to McCarthy's REA model [1982], and Section 9.5 summarizes the chapter. As noted in the footnote earlier, Appendix 9.1 defines the terms used in this chapter.

9.2 The Exchange Events part of the REE model

This section begins with some examples of typical Exchange Events in a trading firm (Section 9.2.1) before presenting a NIAM conceptual schema diagram, Relational Model, and set of Prolog fact types (Section 9.2.2) that define an Exchange Events database more rigorously. To complete the section, Subsection 9.2.3 shows how Exchange Events may be defined for a manufacturing firm.

9.2.1 Example Exchange Events in a Trading Firm

Table 9.1 lists eight Exchange Events (Eno 1 to 8) that illustrate the basic data stored in an REE Exchange Events model. Each Exchange Event has a unique identifier, called an Event number (Eno), a Date (expressed here in Year, Month,
Day-of-month format), an Event type (e.g., "Issue Shares"), and a narration. The resources exchanged are shown like journal entry lines underneath the header information. Each detail line (called a resource-change) shows a line number (Lno), the Asset and Obligation Account being incremented or decremented, the quantity of the resource-change, and wherever necessary (e.g., Eno 2), a money value for the resource-change. There is no need to record the Resource Type involved in each resource-change, since each resource-change affects one, and only one, AO Account, and each AO Account records the number of units of one, and only one, resource type under the enterprise's control. The Resource Type shown in brackets at the end of each detail line in Table 9.1 is for reader information only, it is not part of the record. As with conventional accounting, different AO Accounts are used to record money (or other resources) owed by, or to, different entities, e.g., Accounts Receivable and Payable.

*** Table 9.1 goes approximately here, see next page ***

There is little difference between an REE Exchange Event and a conventional journal entry but the few differences that do exist are critical to the success of the proposed architecture. First, journal entries in any computer-based accounting system, e.g., those reviewed in Chapter 4, normally have a journal entry number, a date, and possibly a narration. REE Exchange Events record the same information, but an Event Type has been added to this "header" information (the top line of each entry in Table 9.1) in each Exchange Event record. The Event Type informs the Interpreter (Chapter 10) of the type of

---

4 For this thesis, the currency unit in which all reports will ultimately be expressed has been assumed to be a dollar.
<table>
<thead>
<tr>
<th>Lno</th>
<th>Account</th>
<th>Quantity Change</th>
<th>Value Change</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bank 1</td>
<td>+100000</td>
<td>($)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Stock Holder 1</td>
<td>-30000</td>
<td>($2 share)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Stock Holder 2</td>
<td>-10000</td>
<td>($2 share)</td>
<td></td>
</tr>
</tbody>
</table>

Eno: 1, Date: 900101, Event type: Issue Shares
(Issue of 40000 $2 shares at a premium.)

| 1   | Inventory 1 | +1000          | +7000        | (Widget 1)            |
| 2   | Inventory 2 | +1000          | +2000        | (Widget 2)            |
| 3   | Supplier 1  | -9000          |             | (Promised $)          |

Eno: 2, Date: 900201, Event type: Credit Purchase
(Purchase 1000 ea. of Widgets 1 & 2 from Supplier 1.)

| 1   | Customer 1  | +5000          |             | (Promised $)          |
| 2   | Inventory 1 | -500           |             | (Widget 1)            |

Eno: 3, Date: 900301, Event type: Credit Sale
(Sale of 500 Widget 1's to Customer 1.)

| 1   | Truck 1     | +1             |             | (Truck 1)             |
| 2   | Supplier 2  | -36500         |             | (Promised $)          |

Eno: 4, Date: 900321, Event type: Credit Purchase
(Purchase of 1990 Ford Delivery Truck from Supplier 2.)

| 1   | Supplier 1  | +4000          |             | (Promised $)          |
| 2   | Bank 1      | -4000          |             | ($)                   |

Eno: 5, Date: 900401, Event type: Pay Supplier
(Pay Supplier 1 $4,000.)

| 1   | Stock Holder 1 - Component | 600 | ($) |
| 2   | Stock Holder 2 - Component | 200 | ($) |
| 3   | Dividend to SH1            | -600| (Promised $) |
| 4   | Dividend to SH2            | -200| (Promised $) |

Eno: 6, Date: 900501, Event type: Declare Dividend
(Declaration of dividend of $0.02 per share.)

| 1   | Office Labour Services     | +200 | (Office Labour Hour) |
| 2   | Bank 1                     | -500 | ($) |

Eno: 7, Date: 900601, Event type: Pay Wages
(Pay office wages of $500 for work in May.)

| 1   | Bank 1                     | -100 | ($) |

Eno: 8, Date: 900701, Event type: Donation
(Donation to Red Cross)

| 1   | Bank 1                     | -100 | ($) |

Table 9.1: Exchange Events in a trading firm

7-8(a)
journal entry it is to generate. Second, the detail lines in any conventional computer-based accounting system will normally have a line number, an account code, and a debit or credit (+/-) amount field. In REE, the accounts are Asset and Obligation Accounts (no revenues or expenses), and the "debits" and "credits" in the column headed "Quantity Change" are in resource units, but otherwise the concepts match exactly. What is new in an REE detail line is an extra column headed "Value" after the Quantity Change column. The purpose of the Value column is explained in the discussion of event number 2 (Eno 2) below.

Other noteworthy characteristics of the Exchange Events in Table 9.1 are as follows:

(a) The first Exchange Event, Eno 1, is a stock/share issue. Forty thousand $2 par value shares have been issued, at a premium of 50 cents each (or thereabouts), to two stockholders in exchange for $100,000 cash. Details of the AO Account, and the number of units of each resource-change, are captured in columns 2 and 3, on separate lines. The plus and minus signs indicate increments and decrements of resources in the specified AO Account. The Interpreter can deduce that the money value of Eno 1, line 1, is $100,000 (because the resource type is monetary), so no Value needs to be defined in column 4 for this line. Similarly, it can also work out the money value of the shares, based on the total number of shares issued.

(b) The second Exchange Event, Eno 2, is a credit purchase. One thousand units of both Widgets 1 and 2 have been exchanged for a promise to Supplier 1 of $9,000. Note that promises like those to Supplier 1 (or from
Customer 1 in Eno 3) are treated as resources. An important piece of information in Eno 2 is Value column after the Quantity Change (+7000 of the cost relates to Lno 1, +2000 to Lno 2). Without this information the unit cost of the two kinds of widgets cannot be determined.

Aside: If the purchase had been effected as two separate exchange transactions, each with only one inventory type per transaction, the first purchase would have been for 7,000 promised dollars, the second for 2,000, and the Value column would not have been needed. In Eno 2 the Value column simply enables a compound Exchange Event to be split into simple two-resource exchanges.

(c) Eno 4, the purchase of a 1990 Ford Delivery Truck, is an old friend from Chapters 7 and 8 (but this time a credit, not cash, purchase). Section 9.3.1 (point (ii)) below describes the way the consumption of resources, both tangible and intangible, is modelled over time.

(d) Eno 6, the declaration of a dividend, is a partial fulfilment of the obligation incurred in Eno 1 when stockholders were promised a stream of dividends if they contributed capital. Although this dividend declaration reduces that initial obligation, it is not possible to record the reduction directly in each Stock Holder account because the number of shares has not changed. In cases like this, where the asset or obligation may be viewed both as a single resource type ("share" in this case) and as an aggregation of one or more resource types (the stream of dividends in this case) it is often helpful to set up component accounts to record the
components as they are removed from (or added to) the initial composite resource. Where the number of components is finite, e.g., the loan discussed in Sections 7.3 and 9.3.3, and components are successively removed the composite resource eventually ceases to exist.

An alternative treatment of a Dividend Declaration would be to record only lines 3 and 4. Any accountant confronted with lines 3 and 4 could prepare the appropriate journal entry. The merits of the two alternatives are discussed briefly in the next chapter, Section 10.1, "Processing of Eno 6".

(e) By the event date of June 1, 1990, the labour service resource acquired in Eno 7 has already been consumed. Again, Section 9.3.1 (point (ii)) below describes the way the consumption of resources, both tangible and intangible, is modelled over time.

(f) Eno 8, the donation, is an example of what Ijiri called a "degenerate" exchange (1975, p.73): something is "exchanged" for nothing. The person who approved the donation must have had some benefit in mind, but there is no readily identifiable benefit to the firm.

9.2.2 A Conceptual Schema for the EE part of the REE model

*** Figures 9.2, 9.3, and 9.4 go approximately here, see next page ***
Figure 9.2: NIAM conceptual schema for the Exchange Events part of the REE accounting model.
Figure 9.3: A Useful Classification Scheme (Resource Type Hierarchy) for Resources
EXCHANGE EVENT (ENO, DATE, EVENT_TYPE, NARRATION)
   (, DOC_NO, RECORDED_BY, DATE_TIME_RECORDED, etc. if required)

RESOURCE_CHANGE (ENO, LNO, AO_ACCOUNT, QUANTITY_CHANGE, VALUE)

AO_ACCOUNT (AO_ACCOUNT, CLASS, RESOURCE, AGENT_ID)
   (, QUANTITY_BALANCE if required)

AO_ACCOUNT_HIERARCHY (PARENT_CLASS, CHILD_CLASS, CHILD_SEQUENCE)

RESOURCE (RESOURCE, RESOURCE_TYPE, UNIT_OF_MEASURE,
   PRICE_INDEX_SERIES)

RESOURCE_TYPE_HIERARCHY (PARENT_RTYPE, CHILD_RTYPE,
   CHILD_RTYPE_SEQUENCE).

EVENT_TYPE_HIERARCHY (PARENT_ETYPE, CHILD_ETYPE).

AGENT (AGENT_ID, ADDRESS)

Figure 9.4: The basic Relational Tables for Resources, Exchange
   Events, and Asset and Obligation (AO) Accounts
Figures 9.2 and 9.3 are NIAM diagrams [Nijssen and Halpin, 1989] of the Exchange Events part of the REE model. A Relational Model schema corresponding to Figure 9.2, 9.3, and two more hierarchies similar to Figure 9.3 (for AO Account Classes and Event Types) is shown in Figure 9.4. As in the previous chapter, relational tables constructed by combining elementary tables from NIAM (on their common keys) should be in Optimal Normal Form [Nijssen and Halpin, 1989]. Prolog fact types based on Figure 9.4's relational database tables are presented in Figure 9.8, in Section 9.5.

At this point it seems worth re-emphasizing that the contents of an REE model are determined solely by the data needs of its Interpreter. To gain an understanding of those needs, first-time readers of this chapter would probably find it beneficial to read Section 10.1 in the next chapter. Section 10.1 contains a detailed, 13-page description of the way a simple Interpreter might use the information in the eight Exchange Events from Table 9.1 to generate historical cost Formula Accounting journal entries. The example helps to explain why the information in the tables in Figure 9.4 is required, and why certain additional information (discussed below) is also needed.

---

5 An E-R model of the same schema is presented in Figure 9.7(b), in Section 9.4 below.

6 This guideline assumes understanding is an iterative process. For the first reading it would probably be better if Sections 9.2.1 and 10.1 had been placed together in a chapter introducing both the notion of an REE model and its Interpreter. However, after the first reading the present chapter structure is probably more useful.
9.2.3 Example Exchange Events in a Manufacturing Firm

In addition to modelling events in a trading firm, the REE model must also be able to model economic events in any other business. In this section, a manufacturing firm is considered. The distinguishing characteristic of a manufacturing activity is that some resources are transformed into other resources, i.e., the enterprise forgoes control over some resources in order to obtain control over others. The problem with recording Production Exchange Events is that precise details of inputs and outputs can often not be known until after production is complete. This can take weeks or months, and many firms will want accounting reports before the whole job, or process, is complete. To provide this information it is convenient to focus on the different cost centres within a firm, e.g., departments, and to treat exchanges of control between cost centres as Exchange Events.

*** Table 9.2 goes approximately here, see next two pages ***

For example, in Table 9.2 a job costing application relating to the production of plastic bottles in the Moulding department of a plastic containers factory has been modelled. Goods move through a number of production steps before completion. Asset and Obligation Accounts have been set up for each class of resource input to, and output from, each production step. Table 9.2 shows eleven Exchange Events. Note that only the first and the last of these Exchange Events have values in the Value column. By maintaining records of values of raw materials, labour, and other resources input into a productive process, an accountant can assign values to the other events. For example, the Interpreter
<table>
<thead>
<tr>
<th>L no</th>
<th>AO Account</th>
<th>Quantity</th>
<th>Value</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RM Inventory 1</td>
<td>500</td>
<td>20100</td>
<td>(20 kg bag poly pellets)</td>
</tr>
<tr>
<td>2</td>
<td>RM Inventory 2</td>
<td>2000</td>
<td>440</td>
<td>(cardboard carton)</td>
</tr>
<tr>
<td>3</td>
<td>Bank 1</td>
<td>-20540</td>
<td></td>
<td>($)</td>
</tr>
</tbody>
</table>

Eno: 1, Date: 900601, Event type: Cash Purchase
(Purchase of Raw Materials)

Eno: 2, Date: 900606, Event type: Raw Materials Issue
(Issue 40 x 20 kg. bags of pellets to Moulding Department)

Eno: 3, Date: 900606, Event type: Record Factory Labour
(79 hours Moulding labour on Job 123)

Eno: 4, Date: 900606, Event type: Interdepartmental Goods Transfer
(Transfer of 10,000 1L bottles from Moulding to Packing)

Eno: 5, Date: 900607, Event type: Raw Materials Issue
(Issue 825 cardboard cartons to Packing Department)

Eno: 6, Date: 900607, Event type: Record Factory Labour
(24 hours Packing labour on Job 123)

Eno: 7, Date: 900607, Event type: Transfer to Finished Goods
(Transfer of 820 cartons of (one dozen) 1L bottles to Finished Goods)

Eno: 8, Date: 900607, Event type: Return to RM Store
(Return 1 x 20 kg. bag of pellets to Raw Material Store)

Table 9.2: Manufacturing Exchange Events (continued ...)

9-13(a)
<table>
<thead>
<tr>
<th>L</th>
<th>AO Account</th>
<th>Quantity Change</th>
<th>Value ($)</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1L bottles Job 123 Step 1</td>
<td>10000</td>
<td></td>
<td>(Unpacked 1L bottle)</td>
</tr>
<tr>
<td>2</td>
<td>RM 1, Job 123, Step1</td>
<td>-39</td>
<td></td>
<td>(20 kg bag poly pellets)</td>
</tr>
<tr>
<td>3</td>
<td>Labour, Job 123, Step 1</td>
<td>-79</td>
<td></td>
<td>(Hour moulding labour)</td>
</tr>
<tr>
<td>4</td>
<td>Carton, 12 x 1L bottles</td>
<td>820</td>
<td></td>
<td>(Carton, 12 x 1L bottles)</td>
</tr>
<tr>
<td>5</td>
<td>1L bottles Job 123 Step 2</td>
<td>-10000</td>
<td></td>
<td>(Unpacked 1L bottle)</td>
</tr>
<tr>
<td>6</td>
<td>RM 2, Job 123, Step 2</td>
<td>-825</td>
<td></td>
<td>(cardboard carton)</td>
</tr>
<tr>
<td>7</td>
<td>Labour, Job 123, Step 2</td>
<td>-24</td>
<td></td>
<td>(Hour packing labour)</td>
</tr>
</tbody>
</table>

**Eno: 9, Date: 900607, Event type: Production Summary**

(Completion of production of Job 123 - Moulding and Packing)

**Eno: 10, Date: 900607, Event type: Recognize Idle Time**

(Foreman’s Report, Idle time, Job 123)

**Eno: 11 Date: 900607, Event type: Record Factory Wages**

(Factory Wages, Employees M, N, and Q, for the last week)

Table 9.2: Manufacturing Exchange Events (continued from previous page)
discussed in Chapter 10 uses standard costing to provide values for HC, CPP, and CC/CPP Formula Accounting journal entries for events 2 through 10.

Eno 2, Table 9.2, is an Exchange Event where the firm gives up control of 40 bags of polyethylene pellets controlled by the Raw Materials department in return for 40 bags of pellets controlled by the Moulding department. Eno 3 shows the Moulding Department acquiring labour resources in return for the obligation to pay wages for 79 hours work. Eno 4 records the transfer of product from the first step in the manufacturing process to the second step, i.e., 1L bottles from the Moulding Department become "raw material" for the Packing Department. Eno 5, 6, and 7 show a similar series of events for the Packing Department: empty cartons are requisitioned from store, labour is used to pack bottles into cartons, and packed cartons are transferred to the next step in the manufacturing process, i.e., to Finished Goods. Eno 8 partially reverses Eno 2 as raw material over-issued is returned to store.

*** Figure 9.5 goes approximately here, see next page ***

Similar events to Events 2-8 can be recorded as often as required until the job is complete. The AO Accounts involved in Job 123, Step 1 (performed by the Moulding Department), are depicted in Figure 9.5. Every AO Account outside the process has a matching AO Account with the same resource type inside the process, e.g., (1) and (2), (3) and (4), and so on. "Production", i.e., conversion of input resources to output resources, takes place as resources "cross" the dotted line from left to right.
REE portrays the resources entering and leaving the process. Conversion occurs as resources "cross" the dotted line.

Pairs of accounts, i.e., (1) and (2), (3) and (4), and (5) and (6), are for the same resource type.

**Figure 9.5: REE View of Manufacturing Process. Job 123, Step 1 (Department M)**
On completion of any job step, all the input-resource AO Accounts will have positive quantity balances, and all the product-resource AO Accounts will have negative quantity balances. At some stage it becomes clear that the job is complete. Job 123's accounts are then closed by recording a Production Summary Exchange Event, e.g., En 9, Table 9.2, that shows the firm relinquishing control of all the input resources in return for all the product resources. The Exchange Event shown zeroes all AO Account balances associated with Job 123, Steps 1 and 2. Only 820 cartons of 12 x 1L bottles (820 x 12 = 9840) have been transferred to Finished Goods, so 160 bottles have been lost during Packing, perhaps due to quality control problems. A resource-change line showing explicitly the loss of the 160 bottles could be added if desired, e.g., if the loss were abnormal, but it is not necessary for variance calculation. For a long job, e.g., one suited to process costing, Production Summary events could be recognized as frequently as desired.

9.2.4 Summary: Exchange Events

Like conventional journal entries, Exchange Events are a powerful way of describing economic events. Much of that power comes from the versatility of Accounts as modelling constructs. In the trading context, Exchange Events are usually multi-resource-type exchanges, where the AO Accounts involved are for different resource types: "our" firm gives up one type of resource in order to obtain control over another. In the case of promised resources, an AO Account also identifies the entity or agent that has promised, or has been promised, the resource(s). However, Events numbered 2 through 8 in the manufacturing example, Table 9.2, involve single-resource-type transfers, where the same
resource type moves from cost-centre to cost-centre. Because AO Accounts model the quantity of one particular resource type controlled by one particular agent they can be used to model either type of exchange with equal ease. In short, because Exchange Events can involve any number of AO Accounts, they are a powerful way of describing economic events. They say a lot, very succinctly.

9.3 Resources in the REE model

As defined earlier, resources are scarce objects that an accounting entity would like to place under its control. It is people, not nature, who impose classifications schemes on resources. A good classification scheme is therefore one that is more useful for a given task than any of the alternatives available. It is also people, not nature, who assign values to those resources. By observing exchanges of resources (often in markets) accountants seek to discover the values that people have assigned to resources. The Exchange Events described in Section 9.2 provide an anchor-point for valuation on the date of exchange, but as time passes values change, and many resources change (physical) form as well. The Resources part of the REE model endeavours to describe the changes in resources and their values that are relevant to the generation of Formula Accounting journal entries.

Recall, the table definition for an AO Account in Figure 9.4 is as follows:

\[ \text{AO\_ACCOUNT}(\text{AO\_ACCOUNT}, \text{CLASS}, \text{RESOURCE}, \text{AGENT\_ID}) \]

Not all resources have a physical form, e.g., promises do not, but it is often helpful to think of physical resources in the discussion that follows. For this reason, the word "physical" is placed in brackets in appropriate places in this section.
9.3.1 **Time-varying resources and resource values**

In classifying resources where both the (physical) form and value of the resource may change it is convenient to use the following three classes:

(i) **No (physical) form change, but value changes**: Examples of resources that do not change (physical) form, yet where value changes are gold bars, financial instruments like stocks and bonds, and foreign currency. These resources are relatively easy to describe (receipts of dividends, bonuses, and interest, can be modelled as degenerate Exchange Events). Their changing values may be modelled with a price index series table like that used in the previous chapter:

```
PRICE_INDEX(INDEX_SERIES, DATE, INDEX_VALUE)
```

(ii) **(Physical) form changes and value changes, but the resource is still considered to be the "same"**: Examples of resources that are usually considered the same, though their (physical) form changes are machines and buildings, which slowly wear out or become obsolescent; services like building occupancy rights and insurance cover, which are used up with the passage of time; and mature livestock, which also "wear out" and die. As discussed in Section 7.2.1(b), accountants have two ways to model changes in value of resources that are considered "the same". One is to index the value of "new" assets and use accounting depreciation to model the consumption of asset services (lines (7.6.4) through (7.6.9)), the other is to model the changing value of the present (used) asset (lines (7.7.1)
through (7.7.4)). The latter case is usually called amortization, e.g., amortization of prepaid insurance. In either case, it is usually necessary to record expectations of asset lives and patterns of consumption of service potential, so that appropriate Formula Accounting journal entries can be generated. A method for defining Expected-value information for resources of this type is presented later in this section.

(iii) (Physical) form change is so great that reclassification of the resource type(s) is required: Examples include the manufacturing events in Section 9.2.2, fashion garments becoming out-of-date, and seedlings becoming trees. These changes can be modelled as exchange "events". If values change along with (physical) form, the new values may be recorded along-side the new resource type in the Exchange Event record.

Ijiri's approach to resource classification [1975, p.101] does not allow the second option above. His resource classification scheme is a simple type (i) or (iii) indifference test. According to Ijiri, if a user is indifferent between two resources they belong to the same resource class and may be meaningfully counted, otherwise they are different. Thus as time passes Ijiri [1975, p.101] finds himself having to reclassify three-year-old machines as four-year-old machines: users are not indifferent to machine age. However, Ijiri's indifference test is too strong. Having to reclassify hundreds, thousands, or more, resources in this way every period seems a pointless exercise. A better approach, that accountants discovered many years ago, is to use a time-invariant type-(ii) classification for some resources. For example, "1990 Ford Delivery Truck" will serve to describe the truck for years.
Thus for type-(ii) resources, the REE model allows the user to define Expected-value functions to specify how, in a world with no inflation, type-(ii) resource values are expected to change over time. Because of uncertainty about the future, accountants tend to use very simple functions for Expected-value functions. For example, straight-line depreciation is based on an Expected-value function determined solely by two parameters: the asset's expected life, and its residual value as a percentage of initial cost\(^9\). Conceptually there is no limit to the functions that may be defined.

In cases like Eno 7, Table 9.1, where the service has been consumed by the time the Exchange Event is recorded, the graph of an Expected-value function for the resource is simply a straight-line that drops from the initial value at time \(t = 0\) to zero for all \(t > 0\). Given this function it is simple to generate journal entries that expense the "asset" in the current period.

An Expected-value function is, of course, just a multi-variable formula with time as one variable. It may therefore be represented using techniques like those used in the last chapter. Two approaches for defining Expected-value functions suggest themselves. The more compact is to use pre-defined functions such as "Straight line" for depreciation. Here one need only indicate the expected life of the asset and its residual value (as a percentage of the initial value) to define the Expected-value of a resource over its life. A relational

---

9 Accountants who argue that depreciation is just an allocation of cost, not a valuation function, tend to forget that they always have to predict future asset lives and residual values, however crudely, before depreciation may be calculated. Inclusion of expectations about the future in the accounting system simply cannot be avoided.
table to link a Resource to its Expected-value function, and the parameters necessary to evaluate the function, is as follows:

```
EXPECTED_VALUE(RESOURCE, EXPECT_VALUE_FUNCTION_NAME, PARAM_1, PARAM_2, ..., PARAM_n)
```

The alternative, more general, way to define an Expected-value function would be to "draw a graph". Points on the graph could be represented by a table like that used for price index series, i.e.,

```
EXPECTED_VALUE(RESOURCE, DATE, EXPECTED_VALUE)
```

In the prototype system developed for this thesis only the former functions, e.g., "Straight line" and "Reducing balance", have been used. However, the idea of a generalized graphical interface for defining Expected-value functions is both appealing and reasonably simple to implement.

### 9.3.2 Composite Resources

Many resources may be viewed as collections of other resources. For example, a mining firm might spend $20 million to build a town near some remote mineral deposit. At that stage it might view the whole town as one resource. Later it might wish to treat parts of the town as separate resources, e.g., to sell houses to employees. A relational table (like a Bill of Materials) to allow a composite resource to be viewed as both one resource, and many component resources (which may in turn have components), is as follows:
9.3.3 Future and Conditional Resources as time-aggregated composite resources

Under Ijiri's scheme [1975, pp.66-67], future assets may exist\textsuperscript{10} but future resources do not. As with Ijiri's classification of depreciable resources (Section 9.3.1), this is not necessarily the best method of resource classification for storing information for generating Formula Accounting journal entries. In the REE model, it is convenient to think of a promise to do something as a resource, e.g., "Promised $" in Table 9.1, Section 9.2, is a resource. This makes it simple to describe a sale on credit as a resource exchange: a resource of type Inventory is given up, a resource of type Promised Money is received. McCarthy's treatment of future assets and resources, which is different, again, is discussed in Section 9.4 below.

A promised resource is similar to, but not the same as, the resource itself. For example, a "Promised "$ is a promise made by one party, to transfer control of another type of resource, a dollar, to another party, at some time in the future. When valuing promises, the extra risk and need for discounting need to be taken into account. In REE, the link between the promise and the resource(s) promised may be modelled as a time-aggregated composite resource. This information, which is illustrated in Figure 9.6, can be used by the Interpreter

\textsuperscript{10} Ijiri's definitions are reproduced as Definitions 9 and 10 in Appendix 9.1.
when valuing a promised resource. The example table in Figure 9.6 defines two types of promised resource. The first line shows a Trade Debt to be a resource where a dollar is to be paid one month after the initiating Exchange Event\textsuperscript{11}. The next three lines define a two year, 15%, loan of unspecified risk; the third line being repayment of principal. In the case of loans-as-obligations, the risk is irrelevant to the firm doing the borrowing. In the case of loans-as-assets, a risk rating may be defined in a separate table (not shown) for each class of security for each borrower.

*** Figure 9.6 goes approximately here, see next page ***

9.3.4 Resource description for a manufacturing firm

*** Figure 9.7 goes approximately here, see next page ***

Figure 9.7 shows five tables that describe the conversion of resources in the job-cost manufacturing environment described in Section 9.2.2. The first two tables define the Asset and Obligation accounts used to record actual quantities of resources consumed and produced in each production step. For example, these tables would define JOB\_NO and STEP\_NO for the three accounts inside the Production box in Figure 9.5 as "123" and "1", respectively.

\textsuperscript{11} Recording time in months seems precise enough for future resource valuation. For example, the term "30 days" in relation to accounts payable is much less precise than the number "30" suggests.
### PROMISED RESOURCE (Promised Resource, Component Number, Component Resource, Months From Event Till Control Transfer, Quantity Of Component)

<table>
<thead>
<tr>
<th>Promised Resource</th>
<th>Component Number</th>
<th>Component Resource</th>
<th>Months From Event Till Control Transfer</th>
<th>Quantity Of Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trade Debt</td>
<td>1</td>
<td>$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Loan 1</td>
<td>1</td>
<td>$</td>
<td>12</td>
<td>0.15</td>
</tr>
<tr>
<td>Loan 1</td>
<td>2</td>
<td>$</td>
<td>24</td>
<td>0.15</td>
</tr>
<tr>
<td>Loan 1</td>
<td>3</td>
<td>$</td>
<td>24</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Figure 9.6: Promised Resources as time-aggregates of other resources**
REE Model:

PRODUCTION_STEP_INPUT_AOAC (AO_ACCOUNT, JOB_NO, STEP_NO)

PRODUCTION_STEP_OUTPUT_AOAC (AO_ACCOUNT, JOB_NO, STEP_NO)

* PRODUCTION_STEP (JOB_NO, STEP_NO, DEPARTMENT, SCHEDULED_START_DATE, SCHEDULED_SET_UP_TIME, SCHEDULED_DURATION)

* CONVERSION_PROCESS_COMPONENTS (PROCESS_NO, COMPONENTRESOURCE, DATE_SET, QUANTITY_OF_COMPONENT_INTO_PROCESS)

* CONVERSION_PROCESS_PRODUCTS (PROCESS_NO, PRODUCT_RESOURCE, DATE_SET, QUANTITY_OF_PRODUCT_FROM_PROCESS, PERCENTAGE_COST_TO_THIS_RESOURCE)

* Not tested in the prototype Interpreter. To simplify calculation of derived standard costs, table COMPOSITE_RESOURCE (see earlier this chapter) was used in place of the two CONVERSION_PROCESS tables above.

Figure 9.7: Resource Conversion Tables for a Manufacturing firm
The remaining three tables have not been used in the prototype Interpreter, but would be used in a more sophisticated program. PRODUCTION_STEP defines set-up times, etc. and identifies the department responsible for controlling each scheduled production step. The final two tables allow manufacturing standards to be defined in resource terms for multiple inputs and multiple outputs from each conversion process. This is a generalization of the COMPOSITERESOURCE table presented in Section 9.3.2 and used in the prototype Interpreter. Such a generalization is necessary because it cannot always be assumed that there will only be one output resource from each production step or process. CONVERSION_PROCESS_COMPONENTS describes the standard resource inputs to a manufacturing step or process. CONVERSION_PROCESSPRODUCTS describes the standard outputs from the process. For joint products, the Interpreter must be told what proportions of input costs are to be allocated to the resources output from the conversion process. The last column of the last table, PERCENTAGE_COST_TO_THIS_Resource, provides rudimentary information for cost allocation, but more complex formulae might be preferred/necessary in some manufacturing processes. In the prototype Interpreter all 100% of Production Step costs are attributed to the single product resource type of each production step.

9.4 Comparison of REE and REA accounting models

There are no absolutes in model building. For a particular task, one model is to be preferred over another if (i) it contains necessary information that cannot be

---

12 These two tables could show units per day, or units per run, or any other convenient metric.
added easily to another model, and/or (ii) it provides a simpler way of doing a job that both models do adequately. Both the REE model and McCarthy's REA model [1982, 1984] are intended to provide high-level, normative descriptions of events in the real world. Neither can be described as "complete" in any absolute sense, since completeness is only meaningful with reference to some specific goal. REE was developed partly because there was information missing in REA, and partly because it seemed a better way to arrange the data for the Interpreter. In the testing described in Chapter 10, if information needed by the Interpreter was missing, REE was modified to store the additional facts. By contrast, McCarthy [1982] developed his REA meta-model without a specific application in mind so it is no surprise that some information needed by the Interpreter is missing from REA.

The two major differences between REE and REA are first, that REE uses accounts as a basic modelling construct and so avoids REA's duality concept, and second, REE contains much more information describing resources, their components, their lives, and changing values, than REA.

9.4.1 Duality vs. Accounts

The first major source of difference between REE and REA is that McCarthy decided not to treat promises, or claims as he calls them, as resources (they are recorded in his system all right, but they are not called resources). Instead, he relies on the potentially troublesome concept discussed in Section 5.3.2 that he calls "duality" relationships to link events. As shown in Section 5.3.2, such duality relationships, or causal chains, can get hopelessly tangled. When this
seems likely to happen, it is easier to break all the links between events by
defining a stock of some intermediate resource type. Stocks of resources are
recorded in accounts. By using the notion of an account, which has a long
pedigree in accounting circles, REE avoids having to rely on duality
relationships to model concepts like accounts receivable.

In addition, there may be a very practical reason for avoiding causal chains in a
Figure 9.1 architecture. It may be impossible to find the information necessary
to build McCarthy's causal chains in the subsystem databases. For instance, in
Section 4.1 it was reported that 82% of the firms with packaged general ledger
systems also had accounts receivable systems. It seems safe to assume that all
these systems record invoices, cash receipts, credit notes, etc. from an account
perspective. Furthermore, those systems using balance-forward accounting will
not store causal chain information, and the information would be unobtainable
without modifying the accounts receivable system. The REE model may
therefore be preferred to REA, simply because it can use existing data within
the subsystems, without modification.

McCarthy's assertion: "It is a primary contention of this paper that the semantic
modelling of accounting object systems should not include elements of double-
entry bookkeeping such as debits, credits and accounts" [1982, pp.559-560] is
therefore too strong. REE is an equally powerful semantic data model that
successfully uses the notion of an account as a modelling primitive.
9.4.2 The absence of resource description in REA

The second major difference between REE and REA is that McCarthy [1982] shows very little interest in describing resources and their values. His Figure 10 [p.574] is the only reference to changing asset values in his paper. There he treats depreciation of a building and amortization of advertising services as being related through "macro-level duality" to cash receipt events. As a modelling construct, macro-level duality seems contrived. By contrast with McCarthy's lack of interest in resource description and valuation, resource description is essential in REE. It is used when generating Formula Accounting journal entries for depreciation and amortization, valuing a firm's manufactured goods, and selecting the appropriate index for inflation-accounting. In other words, if REA were to be used as the data source for the Interpreter, resource description information would need to be added to REA.

9.4.3 Comparing conceptual schemata

An alternative way of comparing REE and REA is to examine schemas drawn using the same modelling methodology. Figure 9.8 presents the Exchange Events part of REE and REA using Entity-Relationship Modelling notation. Figure 9.8(a) is copied directly from [McCarthy, 1982, p.564]. Note the duality relationship linking economic events. The "stock-flow" diamond must have some "quantity" attribute (not shown), so that the number of units of Economic Resource entering, or leaving, the firm's control can be recorded. Figure 9.8(b) is a similarly high-level picture of the REE model with Resource, Event, and AO Accounts located in the same relative positions on the page as in Figure 9.2.
Note that the duality relationship has gone. Instead, the REE model uses the concept of an Asset or Obligation Account to capture the same information.

*** Figure 9.8 goes approximately here, see next page ***

9.5 Chapter summary

The Resources and Exchange Events (REE) accounting model is designed for use at the interface between the various subsystem databases and the Interpreter of Figure 9.1. It is intended to provide a standardized view of data in all the firm's various subsystem databases. The view is intended to be minimal, in the sense that all data are supposed to be necessary for generating Formula Accounting journal entries; no superfluous "possibly useful" data have been modelled. REE models would differ from industry to industry, and possibly from firm to firm, because resources and resource-changes differ so greatly from industry to industry. For example, a retailing firm would have no need for tables describing manufacturing processes.

The Exchange Events part of the REE model is a thorough implementation of Ijiri's Multi-dimensional accounting system [1975], complete with "journal entries" (the Exchange Events themselves), "accounts" (the Asset and Obligation accounts), and a more sophisticated notion of "resource". Because of its account-like structure it retains the flexibility of, and compatibility with, conventional double-entry accounting systems. This compatibility simplifies mapping from subsystem databases to REE, as well as from REE onwards to the
Figure 9.8 Comparison of REE and REA using the Entity-Relationship Model

(a): McCarthy's REA Model [1982, p.564, Figure 5]

(b): ERM Exchange Events model

The terminology in Figure 9.8(b) has been chosen to match McCarthy's REA terminology as closely as possible. A mapping from Figure 9.8(b) to that used in the earlier NIAM diagram, Figure 9.2, is as follows:

<table>
<thead>
<tr>
<th>Figure 9.8(b) Terminology</th>
<th>Figure 9.2 Terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECONOMIC RESOURCE</td>
<td>Unit Resource</td>
</tr>
<tr>
<td>ECONOMIC EVENT</td>
<td>Exchange Event</td>
</tr>
<tr>
<td>stock flow</td>
<td>Resource Change</td>
</tr>
<tr>
<td>ASSET/OBLIGATION ACCOUNT</td>
<td>AO Account</td>
</tr>
<tr>
<td>ECONOMIC AGENT</td>
<td>Agent</td>
</tr>
<tr>
<td>ECONOMIC UNIT (not modelled)</td>
<td></td>
</tr>
</tbody>
</table>

9-27(a)
Formula Accounting general ledger. In a Multi-dimensional accounting system, such as REE, the only accounts are for assets and obligations. There are no accounts for revenues and expenses, nor are there any money-amount balances (except for accounts where the resource itself is monetary). Revenue and Expense accounts lie in the domain of the Formula Accounting general ledger system.

In addition to recording details of Exchange Events, the REE accounting model also records information about resources. Examples discussed in Section 9.3 include details of expected values, changes in the (physical) form or resources over time, information about resources as components of other resources, a scheme for modelling resources to be exchanged in the future, and details of resources consumed and produced during manufacturing. For valuation of resources, it is intended that on the date of exchange the Interpreter will use Ijiri's causal double-entry reasoning to value resources acquired at either their money amount, or the value of the resources given up. Thereafter, the value of a resource may be determined by applying a combination of price index series (for calculating holding gains and losses), Expected-value functions (for calculating depreciation and amortization), and (in the case of a manufacturing process) additional arbitrary cost-allocation rules for "valuing" work-in-process and finished products.

Aside: Amortization rules are more important in REE than in conventional accounting, because there are no expense accounts in Multi-dimensional accounting. Thus in REE money spent on, say, rent, initially gives rise to an asset, Occupancy Rights, which is amortized over its life. In conventional
accounting rent is often expensed directly, with a possible accrual entry only as a balance-day adjustment.

The final section of this chapter compared REE and McCarthy's REA as tools for use by the Interpreter. It was shown that REA contains insufficient information about resources for the generation of depreciation and inflation-adjusted journal entries, and that an account-like structure that recognized promises as resources was likely to be both more compatible with subsystem databases, and less complex to manage than the causal-chain approach. The REE model therefore seems more suited to this application than McCarthy's REA.

*** Figure 9.9 goes approximately here, see next page ***

Figure 9.9 gathers together all the relational database tables defined in this chapter and lists them as Prolog fact types. Compared to the Prolog database for Formula Accounting defined in Figure 8.8, it is apparent that the account definition information is similar (predicate ao_accounts in Figure 9.9 plays a similar role to chart_of_accounts in Figure 8.8), and event description is similar (predicate exchange_events plays a similar role to jeh), but the big difference is that the general ledger contains no information about the firm's underlying resources: a general ledger system models values, not the resources themselves. By contrast, more than half the tables in the REE database in Figure 9.9 are devoted to resource description. This description is necessary if the computer program, not a human accountant, is to prepare Formula Accounting journal entries. The rules this program uses are discussed in the next chapter.
Asset and Obligation Accounts

```prolog
ao_account(AO_account, Description, Class, Resource).
   (, Agent_id, Quantity_balance if required)
ao_account_hierarchy(Parent_class, Child_class, Child_sequence).
* tied_ao_account(AO_account1, AO_account2). /* e.g., leased asset */
```

Resources and their Expected Values

```prolog
resource(Resource, Resource_type, Unit_of_measure, Price_index_series).
resource_type_hierarchy(Parent_Rtype, Child_Rtype, Child_Rtype_sequence).
composite_resource(Resource, Component_number, Date_defined,
   Component_resource, Quantity_of_component).
* promised_resource(Resource, Component_number, Date_defined,
   Component_resource, Months_from_event_till_control_transfer,
   Quantity_of_component).
resource_ev_hierarchy(Parent_ev_type, Child_ev_type, Child_ev_sequence).
   /* ev stands for "expected value" */
expected_value(Resource, Expect_value_function_name, Param_1, Param_2,
   ...., Param_n).
price_index(Series, Date, Index_value) /* common to FA database */
```

Exchange Events

```prolog
exchange_event(Eno, Date, Event_type, Narration).
   (, Doc_no, Recorded_by, Date_time_recorded, etc. if required)
resource_change(Eno, Lno, AO_account, Quantity_change, Value).
event_type_hierarchy(Parent_event_type, Child_event_type).
```

Agent

```prolog
agent(Agent_id, Address). /* e.g., customer, supplier, bank, etc. */
```

Example Resource Description Information for a Manufacturing Firm (below)

```prolog
production_step_input_aoc(AO_account, Job_no, Step_no).
production_step_output_aoc(AO_account, Job_no, Step_no).
* production_step(Job_no, Step_no, Department, Scheduled_start_date,
   Scheduled_set_up_time, Scheduled_duration).
* conversion_process_components(Process_no, Component_resource, Date_set,
   Quantity_of_component_into_process).
* conversion_process_products(Process_no, Product_resource, Date_set,
   Quantity_of_product_from_process, Percentage_cost_to_this_resource).

* = not tested in prototype Interpreter
```

Figure 9.9 Prolog Fact-types for the REE model

9-29 (a)
Appendix 9.1: Definition and Commentary on Terminology used in the REE Model

The terminology used in Chapter 9 is derived from many sources, but the two most important are Ijiri [1967, 1975, 1981] and the FASB's ten "elements" of financial statements from its conceptual framework project [SFAC 3, 1980; SFAC 6, 1985]. This Appendix amplifies the brief description of key terms given in the body of Chapter 9.

In some cases more than one definition is given. The intention is not search for conflict. Rather, it is felt that two definitions, together, often give a better description of a concept than either separately. It is often more useful to work with a more complete, if slightly fuzzy, concept, than to focus on one concept to the exclusion of all others.

Scattered amongst the definitions are a number of comments and assumptions that are intended to amplify the definitions. The assumptions are very close to definitions themselves. For instance, Assumption 1 (after Definition 14) says that asset valuation is solely a matter of human judgement (not some innate property of a resource). This assertion has been called an assumption because it is more like one of Moonitz's Postulates (e.g. Postulate B-2) [1961] than a definition. It is intended as a reminder that values are exogenous data, they cannot be determined by a computer program in an accounting system without externally provided information. In a sense, Assumptions 1, 2 and 3 are definitions of the meaning of "asset valuation" for the purpose of the thesis.

Resources and Exchange Events

Definition 1. An entity is a identifiable unit empowered to control resources. [Ijiri, 1975, Chapter 4, page 52]

Definition 2. Resources are objects that an entity would like to place under its control. They have utility and they are scarce. [Ijiri, 1975, Chapter 4, page 52]

Definition 3. The set of all resources under the control of a given entity is its resource set. This changes over time. [Ijiri, 1975, Chapter 4, page 53]

Definition 4. Some resources are "for all purposes substitutable" for other resources. They form an equivalence class of resources. One may talk of the quantity measure for a particular class of resource. [Ijiri, 1975, Chapter 4, pp. 56-59]

Definition 5. An exchange is an action whereby an entity forgoes control over some resources in order to obtain control over other resources. [Ijiri, 1975, Chapter 4, page 61]

Definition 6. "An economic event is interpreted to be an occurrence in the environment of the firm, either internal or external,
which is of economic significance to the decision makers of the firm". [Colantoni et al., 1971, p.91]

**Definition 7.** Exchange Events are both economic events and exchanges. ("Exchange Event" is a term concocted for this thesis.)

**Comments on Resources and Exchange Events (Definitions 1 through 7)**

(i) Definition 2 says that economic resources are things that are desirable to own or control. It was noted in Chapter 2 that most economic resources can be thought of equally validly as either a single resource, or as a sum of parts. The value (see Assumptions 1 - 3 below for the meaning of "value") of the whole resource, like a machine, is often different to the value of the sum of the component parts. Yet because there is no single, correct level at which resources should be aggregated for valuation purposes, resource definition is a matter of human choice. It follows that human observers, i.e., accountants, must define which things in the real world are to be called single resources. They must also define how the component parts are to be described if ever a resource is split up, or how some group of resources is aggregated to make some new resource, e.g., a manufactured product.

Notwithstanding the problems of resource definition, it is conventional in accounting to think of a firm as owning or controlling a number of resources.

(ii) Definition 6 is very broad. It covers all sorts of events, including conversion of raw materials to finished goods, the announcement of the latest consumer price index figures, receipt of a complaint from a customer about poor quality control, or a declaration of War. Definition 7 restricts the set of economic events of interest to those that Ijiri called exchanges. These exchange events involve the exchange of (control of) one or more resources. They include exchanges in production, and "degenerate" exchanges like donations [Ijiri, 1975, p.73], but exclude announcements like changes in tariff policy.

**Assets and Obligations**

Ijiri [1975] extends the notion of a resource set (Definition 3) to include future increases and decreases in resources controlled by an entity. He then defines assets as follows:

**Definition 8.** Present assets are resources presently under the control of the entity. [Ijiri, 1975, p.66]

**Definition 9.** Future positive assets are resources that are expected to be placed under the control of the entity. [Ijiri, 1975, p.66-7]

**Definition 10.** Future negative assets are resources that are expected to be released from control of the entity. [Ijiri, 1975, p.67]
By contrast, SFAC 6 [FASB, 1985, pp.ix-x] defines assets, liabilities, and equity as follows:

**Definition 11.** Assets are probable future economic benefits obtained or controlled by a particular entity as a result of past economic transactions or events.

**Definition 12.** Liabilities are probable future sacrifices of economic benefits arising from present obligations of a particular entity to transfer assets or provide services to other entities in the future as a result of past transactions or events.

**Definition 13.** Equity is the residual interest in the assets of an entity that remains after deducting its liabilities. In a business enterprise, the equity is the ownership interest.

The two sets of definitions are compared in the Comments below. Before examining them, the following definition and assumptions about valuation are important to this thesis.

**Definition 14.** Obligations are either liabilities or (owners') equity.

**Assumption 1:** Asset valuation is wholly a matter of human opinion or judgement about the probable future economic benefits flowing from control of a resource.

**Assumption 2:** All assets may have associated with them a number of different valuation functions, like historical cost, replacement cost, or net realizable value, based on different assumptions about alternative future uses of the asset and different probabilities of outcomes.

**Assumption 3:** Asset values normally change over time.

**Comments on Assets and Obligations, and their Valuation**

(i) Ijiri's assets and future positive assets correspond to SFAC 6 assets; his future negative assets correspond to SFAC 6 liabilities, but might include owners' equity as well.

(ii) Under Ijiri's definition, assets may have physical form. For example, a firm's truck would be an asset. Under the SFAC 6 definition a truck is not an asset, it is the future services flowing from control of the truck which constitute the asset. Both definitions are similar because it is reasonable to assume that if an entity has control of a resource it also has rights to its future economic benefits. However, the valuation rules are different. Ijiri's definition gives little clue as to how assets should be valued. The SFAC 6 definition suggests a discounted present value approach.
There is no commonly used accounting term, with the possible exception of "equity", to refer to both liabilities and owners' equity. Because the FASB has defined "equity" to be equivalent to "owners' equity" some other term is needed for the more general concept. The term "obligations" is used in this thesis because it may be thought to include a firm's obligations to its owners as well as to lenders.

Assumption 1 recognizes that asset valuation\(^{13}\) is wholly a matter of human judgement. Exchange events give accountants important insights into the values placed on the resources by the respective parties at the time of the exchange. The historical cost of a resource is the money value of the resources given up to acquire it. There is wide, but by no means universal, agreement that the value of a resource at the moment of purchase is its historical cost. However, even for those systems of valuation that agree on the value at the moment of purchase, values will diverge from that moment on.

Assumption 2 explicitly recognizes that over the years there have been such fierce arguments in favour of the different approaches, that from a design point of view it seems unwise to take sides. Indeed, there are strong arguments for different bases of asset valuation for different decisions. Thus, notwithstanding Design Principle 1 in Chapter 6, which favours inflation accounting, there is a good case for arguing that general-purpose accounting systems should be designed to support any number of different methods of valuation that the user may want to use.

Assumption 3, which recognizes that asset values normally change over time, also has important consequences for accounting systems design. Even in historical costing systems the rules for inventory valuation, like LIFO and FIFO, and the rules for depreciation, can be viewed as crude attempts to deal with changing values of assets over time. Sometimes proponents of historical cost accounting say that their system does not attempt to value assets: they say that a balance sheet just reports unexpired costs. But this is just playing with words. Proponents of HC may choose to call their kind of value an "unexpired cost", but it is a form of valuation, nonetheless.

Inflation accountants want to record changes in asset values due to changing market prices (both general and specific), technological obsolescence, wear and tear, maintenance, changes in foreign currency exchange rates, changes in market interest rates, and so on. Although asset and liability values may be viewed as continuous functions of time, users will only supply details of values, even price index series, occasionally. The accounting system will have to interpolate and extrapolate for values at other times.

---

\(^{13}\) Valuation of obligations is included under this term.

9 - 34
Chapter 10. The Interpreter

As shown in the centre of Figure 10.1, the Interpreter is a program that generates Formula Accounting journal entries from data in the REE database. To be effective, the Interpreter should "know" enough about the rules of accounting for it to simulate the behaviour of an expert accountant. Interpreters in different industries and countries might therefore be expected to be different -- because the nature of the resources involved and the revenue and expense recognition rules differ. They would also be expected to change over time, as new accounting standards were promulgated and new financial instruments were developed. If the architecture proposed in this thesis were widely accepted, the accountant of the future would be responsible for maintaining the firm's Interpreter, i.e., keeping it up-to-date, ensuring its journal entries were in accordance with accounting standards, defining data input requirements, etc.

Figure 10.1: The Proposed Architecture for Computer-based Accounting
The prototype Interpreter developed for this thesis can generate FA journal entries for Historic Cost, CPP, and CC/CPP Accounting for a range of event types in the trading and manufacturing industries. As soon as it learns of an exchange event in the outside world the program generates and posts a journal entry in the Formula Accounting ledger. As shown in Chapter 8, these journal entries may appear quite complex. Also as shown in Chapter 8, at the end of each accounting period the Formula Accounting system can evaluate all ledger balances and prepare a "trial balance" for accounting income and balance sheet reports. These reports are the objective of the whole exercise.

At the end of this chapter is a 50-page exhibit, Exhibit 10.1, containing output, including trial balances and listings of journal entries, from four separate runs of the prototype Interpreter (which runs in batch mode). Runs 1 and 2 use the Trading Exchange Events data from Chapter 9, Table 9.1. Run 1 interprets those events from an Historical Cost accounting viewpoint. Run 2 presents a CC/CPP accounting view of the same events. Runs 3 and 4 use the Manufacturing Exchange Events data from Chapter 9, Table 9.2, to generate HC and CC/CPP accounting views, respectively, of those manufacturing events (using standard costing).

Complete program listings of the Prolog programs and databases used to generate the four Runs in Exhibit 10.1 are included at the end of this thesis as

---

1 The structure of the Interpreter (Figure 10.4) makes it simple to add Event-handlers for other types of Exchange Event.
Appendices 1 through 6\(^2\). The code in the Appendices is intended to provide evidence (a) that the printed output shown in Exhibit 10.1 was, in fact, generated by a computer program, and (b) that the proposed architecture -- including the notions of Formula Accounting, an REE database, an Interpreter, and Multiview Accounting -- does actually work. It is not, however, intended that the reader should labour through 100-odd pages of Prolog code and comments.

The objective for this chapter, therefore, is to provide an introduction to the underlying structure of an Interpreter program; to show how it is proposed that the problem of generating Formula Accounting journal entries from an REE database should be accomplished; and to provide sufficient understanding of the procedures involved so that the 50-page Exhibit at the end of the chapter may be sampled with a reasonable understanding of the process that generated it.

To provide some initial insights into the problems of writing an Interpreter program, Section 10.1 "walks" through an artificially-simple, high-level, pseudocode program, called Interpreter 1. Data requirements are considered on an event-by-event basis, over 13 pages, without getting bogged-down in actual program code. Section 10.2 discusses problems with, and improvements to, Interpreter 1. Section 10.3 then turns to the real prototype Prolog Interpreter. Here too, rules are discussed at the "accounting concepts" level, not at the programming level; points are illustrated using examples of output journal

\(^2\) A 360KB diskette containing the same code (the six largest files correspond to the six Appendices) is also included with the thesis. The code is written in standard "Clocksin and Mellish" [1981] Prolog, and should run virtually unaltered on most machines.
entries from Exhibit 10.1. The intention is to explain how the journal entries that appear in Exhibit 10.1 were generated. Because there was little discussion of manufacturing events in the last chapter, particular attention is paid to the generation of journal entries for Manufacturing Exchange Events from Table 9.2. Finally, to "round off" the chapter, Section 10.4 presents a consolidated listing of all database "tables" (actually Prolog predicates) used in the prototype system, and Section 10.5 attempts to summarize and evaluate what development of the system has achieved.

10.1 **A Simple Historical-cost Interpreter: Interpreter 1**

*** Figures 10.2 and 10.3 go approximately here, see next pages ***

Figure 10.2 shows a single page of high-level pseudocode that defines a simple Historical-cost-accounting Interpreter called Interpreter 1. It uses the mapping tables in Figure 10.3 to map from Asset and Obligation accounts in the REE database to General Ledger accounts in the FA database. Interpreter 1 should be thought of as being activated as soon as details of an event are recorded in the REE database. Although the processing sequence in the real Interpreter is slightly different, Interpreter 1 is very similar to the processing module in the main loop of the real Interpreter discussed in Section 10.3.

To see how well Interpreter 1 works, and where extra code and data are required, this section takes 13 pages to simulate execution of Interpreter 1 for the eight Exchange Events in Table 9.1. Each REE Exchange Event from Chapter 9 is reproduced, in turn, in a box below, then followed by discussion of a
use causal-double-entry reasoning to deduce values for each resource change for all resources other than non-"Currency of Account" resources given up.

create new journal entry header with date, event number, narration, etc.

if this is a revenue-recognition event

for each Exchange Event resource-change line do
  if this is a resource acquisition
    place a money value $X on the resource acquired
    identify the GL-account to be debited using map_AOac_to_GLac
    debit the resource-acquired account $X
    identify the revenue account to be credited using map_AC_concept_to_GLac
    credit the revenue account for $X
    if expected-value-resource start expected-value recognition end-if
    if inventory-resource record acquisition end-if
  else /* must be a resource given up */
    place a money value $X on the resource given up
    identify the CoGS account to be debited using map_AC_concept_to_GLac
    debit cost of sales $X
    identify the GL-account to be credited using map_AOac_to_GLac
    credit the resource-given-up account for $X
    if expected-value-resource stop expected-value recognition end-if
    if inventory-resource downgrade inventory record end-if
  end-if
end-do

else /* all events other than revenue recognition */

for each Exchange Event resource-change line do
  if this is a resource acquisition
    place a money value $X on the resource acquired
    identify the GL-account to be debited using map_AOac_to_GLac
    debit the resource-acquired account for $X
    if expected-value-resource start expected-value recognition end-if
    if inventory-resource record acquisition end-if
  else /* must be a resource given up */
    place a money value $X on the resource given up
    identify the GL-account to be credited using map_AOac_to_GLac
    credit the resource-given-up account for $X
    if expected-value-resource stop expected-value recognition end-if
    if inventory-resource downgrade inventory record end-if
  end-if
end-do

if there is a difference between total debits and credits for the event

  identify the non-operating gain/loss account via map_AC_concept_to_GLac
  debit(credit) the gain/loss account with the difference

end-if

check journal entry balances

Post journal entry

Note: The Interpreter uses the three tables in Figure 10.3 in executing this code.

Figure 10.2: Interpreter 1 (A Simple Interpreter for Historical-cost Accounting)
Relational Table: map_AOac_to_GLac(AO_account, GL_account)

map_AOac_to_GLac('Bank1', 'Bank').
map_AOac_to_GLac('StockHolder 1', 'Issued Capital').
map_AOac_to_GLac('StockHolder 2', 'Issued Capital').
map_AOac_to_GLac('Inventory1', 'Inventory').
map_AOac_to_GLac('Inventory2', 'Inventory').
map_AOac_to_GLac('Supplier1', 'Accounts Payable').
map_AOac_to_GLac('Customer1', 'Accounts Receivable').
map_AOac_to_GLac('Truck1', '1990 Ford Truck').
map_AOac_to_GLac('StockHolder 1 - Component', 'Retained Earnings').
map_AOac_to_GLac('StockHolder 2 - Component', 'Retained Earnings').
map_AOac_to_GLac('Dividend to SH1', 'Dividend Payable').
map_AOac_to_GLac('Dividend to SH2', 'Dividend Payable').
map_AOac_to_GLac('Office Labour Services', 'Labour Services').

Relational Table: map_AC_concept_to_GLac(Accounting_concept, GL_account)

map_AC_concept_to_GLac('SalesRevenue', 'Revenue').
map_AC_concept_to_GLac('Cost of Sales', 'Cost of Sales').
map_AC_concept_to_GLac('Gain or Loss', 'Gain or Loss').

Note: The purpose of the above table is clearer if one imagines account numbers in the right-hand column.

Relational Table: event_type_hierarchy(Parent_event_type, Child_event_type)

event_type_hierarchy('Revenue Recognition', 'Credit Sale').
event_type_hierarchy('Non-Revenue Recognition', 'Issue Shares').
event_type_hierarchy('Non-Revenue Recognition', 'Credit Purchase').
event_type_hierarchy('Non-Revenue Recognition', 'Pay Supplier').
event_type_hierarchy('Non-Revenue Recognition', 'Declare Dividend').
event_type_hierarchy('Non-Revenue Recognition', 'Pay Wages').
event_type_hierarchy('Non-Revenue Recognition', 'Donation').

Figure 10.3: Interpreter 1's Internal Database
simulated execution of Interpreter 1 (and any relevant comments). Output from these same eight Exchange Events, generated by the real Prolog Interpreter, is presented in Exhibit 10.1 at the end of this chapter in Runs 1 and 2 (see journal entries zero through seven in both runs).

**Processing of Eno 1:**

<table>
<thead>
<tr>
<th>Line</th>
<th>AO Account</th>
<th>Quantity</th>
<th>Value</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bank 1</td>
<td>+100000</td>
<td>($)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Stock Holder 1</td>
<td>-30000</td>
<td>($2 share)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Stock Holder 2</td>
<td>-10000</td>
<td>($2 share)</td>
<td></td>
</tr>
</tbody>
</table>

Examining Exchange Event number 1, the causal-double-entry module (invoked by the top two lines of code in Figure 10.2) finds that there are only two Resource Types involved: $ and "$2 share". Using the resource_type_hierarchy information in Figure 9.3 it discovers that one of these is of Resource Type "Currency of Account" (i.e., "Australian dollar" for Australia) and one is not, so valuation of the non-monetary resource is possible. The causal-double-entry module therefore decides that in this event 40,000 shares (non-monetary resources) were given up in exchange for $100,000 (monetary resource). It computes that each share has a money-value of $2.50.

Using the event_type_hierarchy table in Figure 10.3 it is determined that this Exchange Event is not a revenue-recognition event. Thus the code in the second

---

3 Table names used in this chapter are those given in the Prolog REE schema, Figure 9.9. For consistency throughout the chapter, all tables referenced in this chapter are in defined in Prolog.
box in Figure 10.2 will be executed. Using the mapping tables in Figure 10.3, the following journal entry results:

<table>
<thead>
<tr>
<th>Dr</th>
<th>Bank</th>
<th>$100,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cr</td>
<td>Issued Capital</td>
<td>$75,000</td>
</tr>
<tr>
<td>Cr</td>
<td>Issued Capital</td>
<td>$25,000</td>
</tr>
</tbody>
</table>

Since neither a $2 share nor a $ (dollar) is an Expected-value resource (found by checking the resource_ev_hierarchy table, where "ev" stands for "expected value") no depreciation or similar entries are required. Further, since neither resource is of type Inventory (found by checking the resource_type_hierarchy, Figure 9.3), the Interpreter's internal record of inventory will not need to be amended. Finally, after processing all three resource-change lines the Interpreter will leave the second box of Figure 10.2, ensure the journal entry balances, and post it to the ledger. Eno 1 is now processed.

Discussion of Processing of Eno 1:

The above journal entry is wrong. Australian company law requires that the difference between the par value of a share ($2 in this case) and the amount that stockholders contribute per share ($2.50 in this case) is recorded in a Share Premium account. Evidently before Interpreter 1 executes the line

credit the resource-given-up account for $X

there should be a test to see if the resource-given-up is a share. If so, the appropriate action is to credit two accounts: Issued Capital and Share Premium
Reserve. To do this correctly, the Interpreter needs to know that the value of $2 shares is $2. This will require a special table in the REE database. Two possible tables to provide this information, one more general than the other, are as follows:

\[
\text{par\_value\_of\_share(Resource,Par\_value, Date\_incorporated).}
\]

\[
\text{normal\_value\_of\_resource(Resource,Normal\_value, Date\_determined).}
\]

With a little extra code, and one of the above two new tables, the modified Interpreter should be able to generate correct Historical-cost journal entries for Eno 1.

Processing of Eno 2:

<table>
<thead>
<tr>
<th>Line</th>
<th>AO Account</th>
<th>Quantity</th>
<th>Value</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inventory 1</td>
<td>+1000</td>
<td>+7000</td>
<td>(Widget 1)</td>
</tr>
<tr>
<td>2</td>
<td>Inventory 2</td>
<td>+1000</td>
<td>+2000</td>
<td>(Widget 2)</td>
</tr>
<tr>
<td>3</td>
<td>Supplier 1</td>
<td>-9000</td>
<td></td>
<td>(Promised $)</td>
</tr>
</tbody>
</table>

Processing of Event Number 2 begins again at the top of Figure 10.2. First, the causal-double-entry module finds that there are three resource types involved: "Widget 1", "Widget 2", and "Promised $". One of these ($) is of type "Currency of Account" but the other two are not, so valuation is possible only if the information in the Value column is used. Using that information, each Widget 1 has value $7, and each Widget 2 has value $2.
Using the event_type_hierarchy table (Figure 10.3) it is determined that this is not a revenue-recognition event. Thus the code in the second box in Figure 10.2 will be executed. Using the mapping tables in Figure 10.3, the following journal entry results:

Dr Inventory $7,000  
Dr Inventory $2,000  
Cr Accounts Payable $9,000

Since neither Widgets nor Promised $ are expected-value resources there will be no depreciation or similar entries required. However, since the first two resources are of type Inventory, the Interpreter’s internal record of inventory will need to be amended to record receipt of 1,000 Widget 1’s at $7 each, and 1,000 Widget 2’s at $2 each. Finally, after processing all resource-change lines the Interpreter will leave the second box in Figure 10.2, ensure the journal entry balances, and post it to the ledger. Eno 2 is now processed. Furthermore, the journal entry is correct.

Processing of Eno 3:

<table>
<thead>
<tr>
<th>Line</th>
<th>AO Account</th>
<th>Quantity</th>
<th>Value</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Customer 1</td>
<td>+5000</td>
<td>(Promised $)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Inventory 1</td>
<td>-500</td>
<td>(Widget 1)</td>
<td></td>
</tr>
</tbody>
</table>

The causal-double-entry module finds that there are two resource types involved: "Promised $" and "Widget 1". However, the resource given up (identified by a negative number in the Quantity column) is not of type "Currency of Account", so the accounting system must already have a value for
this resource type in its database. Therefore the causal-double-entry module leaves the value of the Widget 1's undefined.

Using the event_type_hierarchy table (Figure 10.3) it is determined that Eno 3 is a revenue-recognition event. Thus the code in the first box in Figure 10.2 will be executed. To place a money value on the resource given up, the Interpreter will have to consult its internal inventory records and make a flow assumption (LIFO, FIFO, or standard cost). In this case, irrespective of whether LIFO or FIFO is used, it will place a money value of $7 on each Widget 1. Using the mapping tables in Figure 10.3, the following journal entry then results:

<table>
<thead>
<tr>
<th>Dr</th>
<th>Accounts Receivable</th>
<th>$5,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cr</td>
<td>Sales Revenue</td>
<td>$5,000</td>
</tr>
<tr>
<td>Dr</td>
<td>Cost of Goods Sold</td>
<td>$3,500</td>
</tr>
<tr>
<td>Cr</td>
<td>Inventory</td>
<td>$3,500</td>
</tr>
</tbody>
</table>

Since neither a Promised $ nor a Widget 1 is an expected-value resource there will be no depreciation or similar entries are required. However, since the resource given up is of type Inventory, the Interpreter's internal record of inventory will need to be amended to record that 500 Widget 1's were issued at $7 each. Finally, after processing both resource-change lines the Interpreter will leave the first box of Figure 10.2, ensure the journal entry balances, and post it to the ledger. Eno 3 is now processed. This journal entry, too, is correct.

4 In the real Interpreter, the valuation and amending of inventory records is performed simultaneously.
Processing of Eno 4:

<table>
<thead>
<tr>
<th>Line</th>
<th>AO Account</th>
<th>Quantity</th>
<th>Value (Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1990 Ford Truck</td>
<td>+1</td>
<td>(1990 Ford Truck)</td>
</tr>
<tr>
<td>2</td>
<td>Supplier 2</td>
<td>-36500</td>
<td>(Promised $)</td>
</tr>
</tbody>
</table>

The causal-double-entry module finds that there are two resource types involved: "1990 Ford Truck" and "Promised $". One of these is of type "Currency of Account" (A$) so valuation of the non-monetary resource is possible. By deduction, the single truck has value $36,500.

Using the event_type_hierarchy table (Figure 10.3) it is determined that this is not a revenue-recognition event. Thus the code in the second box in Figure 10.2 is executed. Taking the lines from Eno 4 one at a time, Line 1 is processed as follows. First, using map_AOac_to_GLac for AO Account "Truck 1", a debit to general ledger account "1990 Ford Truck" is generated for $36,500:

\[
\text{Dr} \quad 1990 \text{ Ford Truck} \quad \$36,500
\]

Second, the "start expected-value recognition" module retrieves the following data from the REE and FA databases (see Figures 9.9 and 8.8, respectively):

REE: \texttt{resource_ev_hierarchy('Depreciable Resource', 'Delivery Vehicle', _). expected_value('1990 Ford Truck', 'Straight line', 10, 0).}

and so determines that the Truck is an expected-value resource with an expected life of 10 years and an expected residual value of zero. Thus two more journal entry lines are generated:

Dr  Depr. Exp, 1990 Ford Truck  (if d<3652.5,10*d,36500)
Cr  Accum. Depr, 1990 Ford Truck  (if d<3652.5,10*d,36500)

Continuing with the code in the second box, still for line 1 of Eno 4, the "1990 Ford Truck" is not of type Inventory, so the Interpreter's internal record of inventory need not be updated, and processing for resource-change line 1 is complete. Resource-change line 2 is straight-forward, with no expected-value or inventory processing. It results in a single journal entry line (via the map_AOac_to_GLactable to "Accounts Payable":

Cr  Accounts Payable  $36,500

Finally, having processed both resource-change lines, the Interpreter leaves the second box in Figure 10.2, ensures the journal entry, which is now as follows:

Dr  1990 Ford Truck, at cost  $36,500
Dr  Depr. Exp, 1990 Ford Truck  (if d<3652.5,10*d,36500)
Cr  Accum. Depr, 1990 Ford Truck  (if d<3652.5,10*d,36500)
Cr  Accounts Payable  $36,500

balances, and posts it to the ledger. Eno 4 is now processed, and the journal entry is correct.
Processing of Eno 5:

<table>
<thead>
<tr>
<th>Line</th>
<th>AO Account</th>
<th>Quantity</th>
<th>Value</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Supplier 1</td>
<td>+4000</td>
<td>4000</td>
<td>(Promised $)</td>
</tr>
<tr>
<td>2</td>
<td>Bank 1</td>
<td>-4000</td>
<td>4000</td>
<td>($)</td>
</tr>
</tbody>
</table>

The causal-double-entry module finds that there are two resource types involved: "Promised $" and $. Both of these are of type "Currency of Account" (A$), so no further value processing is required.

Using the event_type_hierarchy table (Figure 10.3) it is determined that this is not a revenue-recognition event. Thus the code in the second box in Figure 10.2 will be executed. Using the mapping tables in Figure 10.3, the following journal entry results:

\[
\begin{align*}
\text{Dr} & \quad \text{Accounts Payable} & \quad 4,000 \\
\text{Cr} & \quad \text{Bank} & \quad 4,000
\end{align*}
\]

Since neither a Promised $ nor a $ (dollar) is an expected-value resource there will be no depreciation or similar entries required. Further, since neither resource is of type Inventory, the Interpreter's internal record of inventory will not need to be updated. Finally, after processing both resource-change lines the Interpreter will leave the second box, ensure the journal entry balances, and post it to the ledger. Eno 5 is now processed, and the journal entry is correct.
Comment on Processing of Eno 5:

Event number 5 is representative of all Accounts Payable transactions. Because of the symmetry of the code in Interpreter 1, one can see that Accounts Receivable events would be processed equally correctly.

Processing of Eno 6:

<table>
<thead>
<tr>
<th>Line</th>
<th>AO Account</th>
<th>Quantity</th>
<th>Value</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stock Holder 1 - Component</td>
<td>+600</td>
<td>($)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Stock Holder 2 - Component</td>
<td>+200</td>
<td>($)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Dividend to SH1</td>
<td>-600</td>
<td>(Promised $)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Dividend to SH2</td>
<td>-200</td>
<td>(Promised $)</td>
<td></td>
</tr>
</tbody>
</table>

The causal-double-entry module finds that there are two resource types involved: "Promised $" and $. Both of these are of type "Currency of Account" (A$), so no further value processing is required.

Using the event_type_hierarchy table (Figure 10.3) it is determined that this is not a revenue-recognition event. Thus the code in the second box in Figure 10.2 will be executed. Using the mapping tables in Figure 10.3, the following journal entry results:

\[
\begin{align*}
\text{Dr} & \quad \text{Retained Earnings} & \quad $600 \\
\text{Dr} & \quad \text{Retained Earnings} & \quad $200 \\
\text{Cr} & \quad \text{Dividend Payable} & \quad $600 \\
\text{Cr} & \quad \text{Dividend Payable} & \quad $200 \\
\end{align*}
\]

Since neither a Promised $ nor a $ (dollar) is an expected-value resource there will be no depreciation or similar entries required. Further, since neither resource is of type Inventory, the Interpreter's internal record of inventory will
not need to be updated. Finally, after processing all four resource-change lines the Interpreter will leave the second box, ensure the journal entry balances, and post it to the ledger. Eno 6 is now processed. Because of the (questionable) description of the Exchange Event with mappings to Retained Earnings via the map_{AOac_to_GLac} table, the journal entry is correct. (Without resource-change lines 1 and 2, some extra code would have to be added to the Interpreter to trap Declare Dividend events, and generate a debit journal entry line or lines to Retained Earnings. The merits of this approach are considered in Section 10.2.)

Processing of Eno 7:

<table>
<thead>
<tr>
<th>Line</th>
<th>AO Account</th>
<th>Quantity</th>
<th>Value</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Office Labour Services</td>
<td>+200</td>
<td></td>
<td>(Office Labour Hour)</td>
</tr>
<tr>
<td>2</td>
<td>Bank 1</td>
<td>-500</td>
<td></td>
<td>($)</td>
</tr>
</tbody>
</table>

The causal-double-entry module finds that there are two resource types involved: "Office Labour Hour" and $. One of these is the "Currency of Account" (A$) so valuation of the non-monetary resource is possible. By deduction, each unit of Office Labour Services has value $2.50.

Using the event_type_hierarchy table (Figure 10.3) it is determined that this is not a revenue-recognition event. Thus the code in the second box in Figure 10.2 will be executed. Using just the mapping tables in Figure 10.3, the following journal entry would result:
However, as 'Office Labour Hour' is an expected-value resource the code within
the second box also requires generation of journal entry lines to record
expectations about the future value of the asset, Labour Services (just debited
above). Using exactly the same procedures as those used for the Truck in Eno 4,
the following information is retrieved from the REE and FA databases (see
Figures 9.9 and 8.8, respectively):

REE:  resource_ev_hierarchy('Amortizable Resource','Office Labour Hour',_).
      expected_value('Office Labour Hour','Benefits already consumed').

FA:  gl_amortization('Labour Services','Wages Expense').

Since the Interpreter has an 'Amortizable Resource' with expected value
'Benefits already consumed' on its hands, it generates the following journal entry
lines in addition to the two above:

Dr  Wages Expense          $500
Cr  Labour Services        $500

Through this journal entry, the asset, Labour Services, is entirely written off in
the current accounting period.

Continuing with the code in the second box, since neither 'Office Labour Hours'
or 'S' is of type Inventory, the Interpreter's internal record of inventory will not
need to be updated. Finally, after processing both resource-change lines the
Interpreter will leave the second box, ensure the four lines of the journal entry balance, and post them to the ledger. Eno 7 is now processed. Furthermore, although the debit and subsequent credit to asset "Labour Services" is rather unconventional, the overall journal entry is correct.

Comments on Processing of Eno 7:
It would have been possible to define the mapping table in Figure 10.3 as follows:

\[
\text{map\_AOac\_to\_GLac('OfficeLabour\_Services', 'Wages\_Expense').}
\]

and to delete the three REE and FA database facts used above to deduce that Office Labour Hours is an amortizable resources that should be charged to the Wages Expense account. However, it is more logically correct to say that an asset, labour services, was acquired and consumed in the one period. Either accounting treatment could be accommodated with equal ease.

Processing of Eno 8:

<table>
<thead>
<tr>
<th>Line</th>
<th>AO Account</th>
<th>Quantity</th>
<th>Value</th>
<th>(Resource Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bank 1</td>
<td>-100</td>
<td>($)</td>
<td></td>
</tr>
</tbody>
</table>

The causal-double-entry module finds that there is only one resource type involved, $, so no further value processing is required. Using the event_type_hierarchy table (Figure 10.3) it is determined that a Donation is not a revenue-recognition event. Thus the code in the second box in Figure 10.2 will
be executed. Using the mapping tables in Figure 10.3, the following journal entry results:

\[
\begin{array}{ccc}
\text{Cr} & \text{Bank} & \$100 \\
\end{array}
\]

Since a $ (dollar) is not an expected-value resource there will be no depreciation or similar entries required. Further, since a $ is not of type Inventory, the Interpreter's internal record of inventory will not need to be updated. Finally, after processing its one resource-change line the Interpreter will leave the second box, discover that the journal entry does not balance, create a second journal entry line as follows:

\[
\begin{array}{ccc}
\text{Dr} & \text{Gain or Loss} & \$100 \\
\end{array}
\]

and post it to the ledger. En 8 is now processed, incorrectly. Evidently there needs to be either (i) code in the Interpreter to trap Donation events and generate a debit to a Donations Expense account, or (ii) an asset like 'Donation Services' defined in the Exchange Event, which is written off to Donation Expense using the same steps as for Office Labour Services in En 7.

10.2 Enhancing Interpreter 1

Considering its simplicity, Interpreter 1 does a surprisingly good job at basic bookkeeping. It got six out of eight journal entries right for the simple

---

5 A critic might argue that all the hard work was done in organizing the data into REE format, defining the AO Accounts and hierarchy information, and in setting up the mapping tables, so of course the translation to journal entries was easy. In reply,
trading Exchange Events in Table 9.1, but it tended to break down on the special cases. The purpose of this section, Section 10.2, is to consider the merit of the design adopted for Interpreter 1, and its potential for improvements. Section 10.2.1 identifies five design questions. Section 10.2.2 proposes some answers.

10.2.1 Five Design Questions

(a) As an alternative to the "generalist" approach of Interpreter 1, where all exchange events are processed by the same piece of code, it would be possible to write special code for each Event Type or for groups of similar Event Types. For instance, it was suggested in Section 7.5 that a simple way to structure the Interpreter would be to use the journal entries like those in Chapter 7 as templates for different event types, and to fill in blanks in the template using details from the REE database. To what extent should the "generalist" or the "specialist" approaches to generating journal entries be favoured? Note that if possible there should be no restrictions of the resource types that may be exchanged. An accounting system should be able to deal with exchanges of any objects of economic value, and one of the attractions of Interpreter 1 is that it is structured so generally that this is possible.

---

One can only point out that the REE model makes no assumptions whatsoever about income measurement. All postings to revenue and expense accounts are generated in accordance with rules encoded in the Interpreter. This is a non-trivial task.

6 This idea is not unlike Attache’s menu system (Section 4.2.4), although the data would now be coming from the REE database, not a user at a keyboard.
(b) The designer has considerable freedom to achieve his or her ends either by enhancing the database, or by adding extra code to the Interpreter program. For example, in the last example, Eno 8, the error in the journal entry could be corrected either by adding data to REE (an extra resource-change line for a dummy resource with zero expected future value) or by adding code in Interpreter 1 to trap and process Donation events differently to other events. Which is the better way to go? What criteria should be used to decide?

(c) Interpreter 1 makes no attempt to generate journal entries for inflation accounting, one of the primary objectives of this study. How much extra work would be required to enhance Interpreter 1 to generate, say, CC/CPP accounting journal entries for all events?

(d) For the Interpreter to be able to "place a money value on the resource given up" (one of its fundamental modules) it must maintain in its private database or elsewhere:

(i) full LIFO, FIFO, or standard cost inventory records\(^7\), and

(ii) full depreciable- and amortizable-asset records, at the individual item level.

---

\(^7\) If standard costs were used, the Figure 10.2 Interpreter would need to be modified to isolate purchase price variances at the time of resource acquisition.
Both requirements (i) and (ii) mean that the REE model must provide details of goods purchased and sold at the individual item level. Such details are not required for monetary resources, e.g., Cash at Bank, Accounts Receivable, and Accounts Payable, or any resources where money values are determined by current market prices, e.g., Foreign Exchange. Should FA general ledger accounts be maintained for individual inventory items, or for individual assets, or not?

(e) Interpreter 1 seems to handle journal entries for a trading firm quite well, but what sort of problems arise in a different industry, e.g., in a manufacturing firm?

10.2.2 Design Decisions: Answers to the Five Questions

*** Figure 10.4 goes approximately here, see next page ***

The most straight-forward answer to the first and last questions, (a) and (e), is to structure Interpreter 2 as shown in Figure 10.4. In essence, the two boxes containing code in Figure 10.2 have now become part of a large CASE statement with different boxes for different groups of related events. Thus Cash Sale and Credit Sale events might both be handled by the first box, Capital Raising events in the second box, Manufacturing events in the third box, and so on. Events not requiring special treatment fall through to the last box. Wherever possible all boxes would share common core modules, e.g., to place money-values on resources, generate journal entries for depreciation and amortization, maintain inventory records, generate and manipulate formulae for Formula
use causal-double-entry reasoning to deduce values for each resource change for all resources other than non-"Currency of Account" resources given up create new journal entry header with date, event number, narration, etc.

if this is a revenue-recognition event

else if this is a capital-raising event

else if this is a manufacturing event

else /* all other events */

end-if

cHECK journal entry balances
post journal entry

Note: The Interpreter still uses the three tables in Figure 10.3 in executing this code.

Figure 10.4: Structure for a More General Interpreter
Accounting, and so on. Structuring an Interpreter this way partitions the overall problem into many smaller problems without imposing any restrictions on the resource-type exchanges that the Interpreter can handle: anything may still be exchanged for anything.

Under the program structure shown in Figure 10.4, the human accountant using the system could think of each Event Type as being processed by its own module in the Interpreter. He or she could then have a book showing, for each event type, examples of REE data required, and examples of the corresponding FA journal entries generated by the Interpreter.

The decision to structure the Interpreter as shown in Figure 10.4 also partly answers question (b) above. The creation of each new event handler is a sort of learning by the Interpreter: the Interpreter gets "smarter". By contrast, if the accountant specially modifies the way events are encoded in the REE model so that the Interpreter need not be changed, it is the accountant who is adapting, not the Interpreter. It is therefore planned that as a matter of policy, the structure of exchange events in REE should be as simple as possible (and as

There are times when these general-purpose common modules do not do the job and special routines must be written. For example, the mixed dates for index values in the Cost of Sales journal entry in Chapter 7, Figure 7.9 (where subscripts 0 and 1 in journal-entry lines (7.9.11), (7.9.12), and (7.9.13), represent the date of acquisition and date of sale, respectively) meant that a special sub-module was required for the cost of sales side of the Credit Sale journal entry for CC/CPP accounting. When the module for the Credit Sale event was first coded (based, supposedly, on the templates in Chapter 7) the standard modules were used. But the journal entries did not balance. Careful analysis of the problem showed that the standard modules indexed money values from the Sale event date, not the date of acquisition of the inventory, and a special module was required. The moral is that unless a design is actually tested (through implementation) it is all too easy to accept erroneous reasoning as valid.
consistent with normal accounting conventions/rules as possible), and that the Interpreter code should be changed whenever any new, unusual event is encountered. At some time in the future it may even be possible for the Interpreter to reason that a new event is "like" some existing event, and work out for itself how the new event should be handled. For the present, decisions about accounting rules can be made by an accountant and encoded by a programmer. Use of common modules like those in Interpreter 1 to "place a money value on a resource change", and "start expected-value recognition" (e.g., depreciation and amortization) ensures that defining new event handlers is not too difficult a task.\footnote{The easiest way to code a new module for the Interpreter is to copy a similar existing module and modify it.}

This leaves questions (c) and (d) to be addressed. Question (d), placing money values on resources given up, will be considered first. It was noted earlier that for the Interpreter to be able to "place a money value on the resource" it must maintain in its private database or elsewhere:

(i) full LIFO, FIFO, or standard cost inventory records, and

(ii) full depreciable and amortizable asset records, at the individual item level.

Despite the fact that values placed on any asset that has been held by a firm for some time are fraught with estimation error, the rules of accounting are quite strict about how these estimates are to be made. By convention, inventory
items are not depreciated. Where individual items are not specifically identified, e.g., by serial number, flow assumptions like LIFO and FIFO are used to resolve estimation puzzles. For assets whose service potential is consumed in earning income, i.e., depreciable and amortizable assets, a different set of rules must be followed. Generally, depreciable and amortizable assets are individually identified and expense recognition for expected-value changes for these individual assets continues until the asset is sold.

In the proposed architecture it is assumed that details of all these individual assets and asset types will be maintained in special-purpose transaction processing sub-systems such as Inventory Control and Fixed Assets. With all the details "at their fingertips" it would seem easiest if those subsystems prepared the appropriate Formula Accounting journal entries as required (i.e., the system would use the upper path in Figure 10.1). In such a case, the Interpreter program would not need to maintain inventory and "expected-value" records itself. However, for reasons given in Chapter 6, the Interpreter in this thesis is required to generate its own journal entries. It must therefore maintain its own data about inventory and depreciable and amortizable (expected-value) assets.

A simple way to maintain value information for individual depreciable and amortizable assets is to set up individual accounts in the FA general ledger for each asset. In the case of depreciable assets, this means three new accounts: the Asset "at cost", Accumulated Depreciation, and Depreciation Expense, for

---

10 Perishable assets, or assets subject to rapid technological obsolescence are presumably caught by the "lower of cost and market" rule. They could, of course, be revalued continuously using formulae.
each asset. These account balances are simple to aggregate for reporting purposes using the Class information in each account and the glAC_hierarchy table (Figures 8.8 and 8.3). However, it is not sufficient to set up individual general ledger accounts for each different type of inventory because ledger accounts record only balance information; the flow information needed for inventory valuation is lost. So for inventory valuation the Interpreter needs to maintain special information in its own internal database.

The final question posed in Section 10.2.1 but not yet "solved" was (c), the generation of FA journal entries for inflation accounting. Here, the problem partitioning achieved by the Interpreter structure in Figure 10.4 also assists. In essence, each box for each event type in Figure 10.4 must itself contain a CASE statement, with different accounting journal entries raised for different methods of handling inflation. Again, in the prototype system an accountant must define the rules for each case. These are then encoded in the Interpreter by a programmer. In other words, the journal-entry-template approach suggested in Chapter 7, Section 7.5, will be followed; the Interpreter program will not be asked to deduce its own rules.

10.3 The Prototype Interpreter

The prototype Interpreter extends Interpreter 1 in two ways. First, its top level module uses the block structure suggested in Figure 10.4. This is a large CASE statement that directs program execution to different modules for different event types. Second, within each module there is code to deal with three types of inflation accounting, Historical Cost, Current Purchasing Power (CPP), and
Current Cost/Current Purchasing Power (CC/CPP). CC/CPP accounting is both theoretically attractive (Chapter 2, Section 2.4) and reasonably straightforward to implement. Gains or losses relative to the general level of prices are identified on a line-by-line basis for each Exchange Event. All other money values, e.g., in revenue and expense measurement, are indexed for changes in the general level of prices.

*** Table 10.1 goes approximately here, see next page ***

As mentioned in the introduction to this chapter, Prolog code for the entire prototype Formula Accounting/Interpreter system is reproduced in Appendices 1 through 6. An outline of the content of these Appendices is presented in Table 10.1. Each Appendix has an index to all its procedures on the last page. The first page of Appendix 2 provides a bridge between the pseudocode of Interpreter 1 and the actual Prolog code. However, the program code is included mainly as evidence that the output in Exhibit 10.1 was computer generated. Anyone who has followed the examples of Formula Accounting journal entries in Chapter 8, Exhibit 8.1, and the explanation of Interpreter 1 in this Section 10.1 of this chapter should be able to scan through Runs 1 and 2 in Exhibit 10.1 without much trouble.

10.3.1 Run 1, Exhibit 10.1

Output from Run 1, an Historical Cost, FIFO, interpretation of the events in Table 9.1, appears on pages 2 through 13 of Exhibit 10.1. HC journal entries for the eight Exchange Events in Table 9.1 have already been discussed at great
<table>
<thead>
<tr>
<th>Appendix</th>
<th>Lines</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>926</td>
<td>Contains almost all the code for the Formula Accounting system. This code performs the listing, posting, balance-checking, account-closing, and formula-evaluation functions described in Chapter 8 and illustrated in Exhibit 8.1.</td>
</tr>
<tr>
<td>Interpreter</td>
<td>1391</td>
<td>Contains the bulk of the code for the Interpreter. This code initializes the Multiview system, performs causal double-entry reasoning, generates Formula Accounting journal entries (including those for depreciation, amortization, and standard costing variances), maintains inventory records (LIFO, FIFO, and Standard Cost), and contains all the code for the three different variants of inflation accounting. As shown in Interpreter 1, it uses mapping tables like those shown in Figure 10.3 to link together the REE and conventional accounting models.</td>
</tr>
<tr>
<td>Event Handler</td>
<td>504</td>
<td>This code contains nine specific Event handlers for the Interpreter. These are as follows: cash_payment, cash_receipt, credit_sale, donation, issue_shares, manuf_event, open_accounts, production_summary, and sale_depreciable_asset. Some of these modules are quite short, e.g., cash_payment is 15 lines long, because they use code in the &quot;core&quot; Interpreter wherever possible.</td>
</tr>
<tr>
<td>Prolog Library</td>
<td>350</td>
<td>Low-level Prolog procedures like once, sum, max, write_fixed_length, is_a, and date_function, that extend the minimal commands provided by basic &quot;Clocksin and Mellish&quot; Prolog, e.g., there is not easy way to format printed output in Prolog. (Prolog is, however, excellent for manipulating &quot;tables&quot; of data.)</td>
</tr>
<tr>
<td>Trading Database</td>
<td>947</td>
<td>Trading database, containing all database &quot;tables&quot; including the facts for Table 9.1 in lines 616-642. This database was used in Runs 1 and 2 in Exhibit 10.1. Tables are organized in three main groups, being those for Formula Accounting, the REE model (including the Exchange Events), and the Interpreter, respectively.</td>
</tr>
<tr>
<td>Manufacturing Database</td>
<td>882</td>
<td>Manufacturing database, containing all database &quot;tables&quot; including the facts for Table 9.2 in lines 572-625. When used in Runs 3 and 4 of Exhibit 10.1, this database completely replaced the Trading database. Many of the tables in the two databases, e.g., the various object-type hierarchies, are similar.</td>
</tr>
</tbody>
</table>

Table 10.1: Prolog code used for the Prototype Interpreter
length in Section 10.1. Since these are familiar to the reader it seems likely that the fastest way to understand Run 1 would be to begin by comparing the journal entries on pages 8 through 10 to the events in Table 9.1. The Interpreter's task is to generate these journal entries, and to post them to the ledger. Once this objective is clear, one can work quite quickly through the printed output from Run 1.

Output from Run 1 begins on page 2 of Exhibit 10.1. The ed(...) predicates commencing on the last two lines before the footnotes on page 2 are the result of applying causal double-entry valuation to the resource-changes shown in Table 9.1. Page 4 of Exhibit 10.1 shows various event-handlers being called to process the Exchange Events. Pages 5 and 6 show the ledger account balances resulting from posting the journal entries on pages 8 through 10 to the ledger. Page 7, the "Money Trial Balance", provides an easy way for a human to review the state of the ledger (as at 1 January, 1991). Following the listing of journal entries, procedure close_revenue_acs is called to transfer revenue and expense account balances to Profit and Loss. The closing journal entry generated by the Formula Accounting system, journal entry 8, is listed on pages 10 and 11. This is followed by opening trial balance "reports" for the next period, on pages 11 through 13.

10.3.2 Run 2, Exhibit 10.1

Journal entries in Run 2 (Exhibit 10.1, pages 14-27) provide a CC/CPP accounting interpretation of the events in Table 9.1. The sequence of "reports" used in Run 1 is repeated in Run 2. Explanation of the accounting involved for
CC/CPP accounting was given in Chapter 7. Journal entry 3 on page 20 of Exhibit 10.1 (the 1990 Ford Truck in Run 2) is deliberately identical to journal entry 1 of Exhibit 8.1 (only the formula numbers are different).

The basic code for CC/CPP accounting is little different to that used for HC accounting. All that happens is that every time the Interpreter gets to a module that generates a FA journal entry it checks to see what sort of inflation accounting it is supposed to be performing. Every Asset and Obligation Account in REE is for one and only one resource (defined in the ao_account table), and every resource has an associated price-index series (defined in the resource table), so each resource-change is associated with one and only one price-index series. The algorithm used by the Interpreter as it is about to generate a journal entry line is as follows:

```plaintext
if the resource-change is monetary
    debit or credit the GL account for a fixed money-amount
    debit or credit the Gearing Gain or Loss account for a formula based on the general price index, g
else
    debit or credit the GL account to index the money-amount by the appropriate specific price index
    debit or credit the difference between the specific price index figure and the general price index figure to a Real Holding Gain account called "CCA Gain above General Prices".
```

CPP accounting uses a similar algorithm, but uses the general price index series for all non-monetary resources. Index values between the dates specified in the price_index table are deduced by linear interpolation and extrapolation.

---

11 Foreign currency is not.
The effect of the above algorithm is visible in the journal entries on pages 19 through 23 of Exhibit 10.1. For instance, in journal entry zero, page 19, the debit to Bank in line 1 is followed immediately by an entry to Gearing Gain, on line 2. Share-holders funds in lines 3 through 6 are indexed at the general level of prices, $g$, so no Gain or Loss entries are required.

In journal entry 1, page 19, the Widgets are indexed by the specific price index for inventory, $s_i$, on lines 1 and 3, and real holding gains are recognized on lines 2 and 4.

Much the same approach is adopted for each resource-change line in the eight Exchange Events modelled by the journal entries in pages 19-22 of Exhibit 10.1. Monetary changes give rise to Gearing Gains. Non-monetary changes give rise to Real Holding Gains. All revenue and expense journal entries are indexed by the general price level variable, $g$.

10.3.3 Run 3, Exhibit 10.1

Runs 3 and 4 in Exhibit 10.1 are for the Manufacturing Exchange Events in Table 9.2. The journal entries required for an accounting interpretation of these Exchange Events have not been discussed earlier, and warrant further explanation. The basic situation is depicted in Figure 10.5.

*** Figure 10.5 goes approximately here, see next page ***
Conventional Accounting portrays the COSTS of resources entering and leaving the process. Conversion occurs as resources "cross" the dotted line.

Conventional-Accounting View:

<table>
<thead>
<tr>
<th>Input AO Accounts</th>
<th>Output AO Accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Raw Material 1</td>
<td>Intermediate Product 1</td>
</tr>
<tr>
<td>Factory Labour Services</td>
<td>xx</td>
</tr>
<tr>
<td>(3) Labour in Process</td>
<td>xx</td>
</tr>
<tr>
<td>(4) Machine Services</td>
<td>xx</td>
</tr>
<tr>
<td>Set-up Labour Services</td>
<td>Variance</td>
</tr>
<tr>
<td>Valued at standard cost</td>
<td>Account 1</td>
</tr>
<tr>
<td>(5) Intermediate Product 1</td>
<td></td>
</tr>
<tr>
<td>(6) Finished Good 1</td>
<td>Matl. + Labour + Overhead COSTS</td>
</tr>
</tbody>
</table>

REE portrays the resources entering and leaving the process. Conversion occurs as resources "cross" the dotted line.

REE View:

<table>
<thead>
<tr>
<th>Input AO Accounts</th>
<th>Output AO Accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Raw Matl. 1</td>
<td>Intermediate Product 1</td>
</tr>
<tr>
<td>Factory Labour Services</td>
<td>xx</td>
</tr>
<tr>
<td>(3) Labour</td>
<td>xx</td>
</tr>
<tr>
<td>(4) Machine Services</td>
<td>xx</td>
</tr>
<tr>
<td>(5) Intermediate Product 1</td>
<td></td>
</tr>
<tr>
<td>(6) Finished Good 1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10.5: Comparing REE and Conventional Accounting Views of a Manufacturing Process Step
The top of Figure 10.5 is a generalization of Figure 9.5. It shows a production step for some productive process, represented by the box, with REE-model AO Accounts inside the box. Resources of one kind and another flow into the productive process on the left of the box. They are transformed as they cross the dotted line into new resources that flow out, to other productive process steps or to Finished Goods, on the right.

A conventional accounting view of the same production step is shown in the lower half of Figure 10.5. Here, the values of the resources that flow into, and out of, the production step are recorded. What is needed for the Interpreter is a means of mapping from the diagram at the top to the diagram at the bottom. The mapping between accounts outside the boxes is straight-forward: the normal mapping table, map_AOac_to_GLac, may be used. However, the mapping for accounts inside the boxes requires more information.

For the prototype Interpreter, the following table, similar to the two at the top of Figure 9.7, provides the mapping for accounts within the boxes:

\[
\text{production\_step\_GLacs(Job\_no,Step\_no,M\_glac,L\_glac,OH\_incurred\_glac, OH\_applied\_glac)}
\]

The tables in Figure 9.7 identify the Job_no and Step_no for each AO_Account in the upper box in Figure 10.5. The new table, production_step/GLacs, then allows the corresponding conventional-accounting general ledger accounts for Material-in-Process (M_glac), Labour-in-Process (L_glac), and the two Overhead accounts (OH_incurred_glac and OH_applied_glac) in each production step to be
identified. For resources transferred out of the production step, into, say, Finished Goods, a debit is posted to Finished Goods, and credits are posted to the three accounts M_glac, L_glac, and OH_applied_glac. These credits are indicated by the yy's in Figure 10.5.

Run 3 begins on page 28 of Exhibit 10.1. Most of the ed(...) records generated by the causal double-entry module (pp.28-29) have null fields (_1) for the value column because their values are to be determined by the Interpreter. The prototype Interpreter uses standard costing for valuing resource changes for these manufacturing events, standard costs being stored in the following table:

\[
\text{unit\_standard\_cost(\text{Resource,Date\_set},\text{Standard\_cost})}
\]

Standard costs may change, so the Interpreter uses the latest standard cost defined before the event date in question. Standard costs for composite resources, i.e., any manufactured product, can be derived using the usual rules for aggregating material, labour, and overhead costs. Using two more tables:

\[
\text{department\_responsible\_for\_manufacture(\text{Resource,Department})}
\]
\[
\text{departmental\_oh\_rate(\text{Department,Date\_set},\text{OH\_rate,Basis\_of\_variation})}
\]

it is possible to calculate the standard cost components of all composite resources. Results from such a calculation are reported on pages 29 and 30 of Run 3. As shown on those pages they are stored in a table as follows:
Page 30 shows the event-handlers called by the Interpreter as it processes the eleven Exchange Events from Table 9.2. It is worth pausing to examine the journal entries on pages 34-38.

Journal entry zero (page 34) does not correspond to any Exchange Event. It was entered from a key-board and allocates any balance that appears in GL account 2802 as follows:

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr 2810</td>
<td>Overhead Incurred - Moulding</td>
<td>0.6 * '2802'</td>
</tr>
<tr>
<td>Dr 2820</td>
<td>Overhead Incurred - Packing</td>
<td>0.3 * '2802'</td>
</tr>
<tr>
<td>Dr 2830</td>
<td>Office Share of Rent</td>
<td>0.1 * '2802'</td>
</tr>
<tr>
<td>Cr 2803</td>
<td>Production OH - Contra</td>
<td>-1 * '2802'</td>
</tr>
</tbody>
</table>

The effect of these allocations is apparent in the trial balance listings for the above accounts, e.g., on page 32 or 33.

In journal entry 1 (page 34) Prepaid Rent is treated as an asset and amortized over the forthcoming month. If entries like this are going to be created each month it would be helpful if the month-end procedures could analyze conditional balance formulae, and replace formulae that are invariant for all future time by unconditional formulae. (This would save both future processing time and storage space.)
Journal entries 2 through 4 (page 35) describe simple resource-change events. (The event number for each journal entry is shown in the jeh line, field 3.) The Materials in Progress and Labour in Progress accounts debited in journal entries 3 and 4 were determined by a special event handler, called manuf_event, using the two tables discussed in relation to Figure 10.5 (production_step_input_aoac and production_step_GLacs).

Journal entry 5 (page 35) corresponds to a transfer-out of resources from Job 123, Step 1, to Job 123, Step 2. The debit entry uses the derived unit standard cost calculated earlier. The credit entries, and there are three of them, use the derived unit standard costs for values, and the production_step_output_aoac and production_step_GLacs tables to determine the accounts to credit.

Journal entries 6, 7, and 8 (page 36) are similar to journal entries 3, 4, and 5. Likewise, journal entry 9 is a simple resource change as goods are returned to the Raw Materials store.

Journal entry 10 (pages 36-37) is the response to the Production Summary event, Eno 9, in Table 9.2. The account balances transferred to Profit and Loss are the materials and labour usage variances for the Moulding and Packing Departments for Job 123, Steps 1 and 2 (material price variances were isolated in journal entry 2, page 34). Overhead variances cannot be calculated on a job by job basis because many overhead charges cannot be traced to a particular job. Overhead variances should therefore be calculated periodically, e.g. when revenue and expense accounts are closed. Such periodic transfers have not been coded into
the current version of the prototype Interpreter, but the basic data are present in the database.

Journal entry 11 (page 37) treats Idle time in the two departments as Overhead. Journal entry 12 (page 37) isolates a Labour Rate variance as the wages obligation for the week is recognized.

10.3.4 Run 4, Exhibit 10.1

Run 4 generates a CC/CPP accounting view of the same Exchange Events, i.e., Table 9.2, as Run 3. The most difficult problem in generating CC/CPP accounts for the manufacturing example was deciding how to index standard costs. First, it seemed in the spirit of CCA to index all standard costs by the specific price index series for their resource type. This would mean, for example, that material price variances would be calculated after allowing for trend of price changes for each type of material. However, the decision to index standard costs from Date_set to Event_date creates a problem for variance analysis in a manufacturing industry. If the value-weighted price indexes of resources input to a production process do not match those of resources output from the same process, there will be unavoidable differences in debits and credits to Materials-in-Process and Labour-in-Process accounts during all periods after the date the standard were set.

For example, assume that on the day standards are set it takes $100 of inputs to make one unit of output, so the standard cost of output is $100. If production is exactly in line with the standard, debits and credits to the production step
account will exactly balance. Now assume that in the next few months the price index for inputs rises from 100 to 105, and the price index for the output rises from 100 to 110. Now, if production is exactly in line with standard, the production step account will be credited with $5 more than it is debited for every unit of production. At present these are reported as Materials- and Labour-usage variances, which may technically be wrong. There is no simple solution to this problem. The choices appear to be to (i) choose not to index standards, (ii) reset standards periodically, or (iii) define the index for manufacturing product as a value-weighted sum of input resource price indices. None of these solutions is entirely satisfactory. In the prototype Interpreter, standard costs are indexed, which means that some material and labour usage variances are not strictly correct.

Journal entries for Run 4 appear on pages 44-50 of Exhibit 10.1. Journal entry 0, the cost allocation, need not be indexed (as for Run 3, it was hard-coded). The rent in journal entry 1 was indexed by the general level of prices, g, and the credit to Bank gave rise to Gearing Gains. In journal entry 2 Real Holding Gains must be recognized both on the Raw Materials and the Price Variances (otherwise the journal entry will not balance). These are shown on page 45. For journal entry 3 the line-by-line algorithm recognizes a Real Holding Gain on line 2 and an exactly matching Loss on line 4. This is inefficient, but not wrong. Much the same comment can be made about the remaining nine journal entries. In short, Run 4 shows that the proposed computer-based accounting system architecture generates correct accounting journal entries for inflation accounting in a manufacturing environment.
10.4 The Full Database used by the Interpreter

*** Table 10.2 goes approximately here, see next page ***

Table 10.2 summarizes the database used by the Interpreter. It was prepared by taking the machine-generated index from the last page of the Manufacturing Database, Appendix 6, then rearranging and grouping table names under three headings corresponding to Chapters 8, 9 and 10. There are some small differences in spelling of tables names, e.g., "resource_type_hierarchy" in Figure 9.9 was implemented as "resource_hierarchy" in Table 10.2, but data structures are as described in Chapter 8 through to this point in Chapter 10.

The successful generation of the journal entries in the fifty-page Exhibit 10.1 provides an important test of completeness of the database schema in Table 10.2, and in particular, the REE model specified in Chapter 9. The schema in Table 10.2 is from a tested, working system. Alternative database schemas could be designed, but they would need to contain information very similar to that presently defined in the Table 10.2 database. Table 10.2 is therefore a reasonable guide to the data requirements of future Formula-Accounting-based Multiview Accounting systems.13

Scanning Table 10.2, it is evident that the data requirements for an REE-based Interpreter system are considerably greater than those for a free-standing

---

13 It is only a guide, because it has been designed specifically to cope with the relatively simple Exchange Event types presented in Tables 9.1 and 9.2. Additional data could well be needed for special events not considered in Tables 9.1 and 9.2.
1. **Formula Accounting** (cf. Figure 8.8)
   - chart \_ \_ \_ accounts(GLacno, Description, Class)
   - glAC \_ hierarchy(P \_ seq,P \_ name, C \_ seq, C \_ name)
   - gl \_ depreciation(GL \_ ACno, Accum \_ Depr \_ ACno, Depr \_ Exp \_ ACno)
   - gl \_ amortization(GL \_ ACno, Expense \_ ACno)
   - price \_ index(Series, Date \_ set, Index \_ value)

   Five FA Tables Created by the Interpreter:
   - jeh, jed, formula, uncondit \_ formula, gl \_ ac \_ balance

2. **Resources and Exchange Events Accounting Model** (see Figure 9.9)
   - ao \_ account(ACno, Description, Class, Rno, Agent \_ id, Controller)
   - ao \_ hierarchy(PI, Pname, Cseq, Cname)
   - tied \_ asset \_ obligation(ACno1, ACno2)

   resource(Rno, Description, Unit \_ of \_ measure, Rtype)
   - resource \_ hierarchy(PI, Pname, Cseq, Cname)
   - composite \_ resource(Composite \_ Rno, Date \_ defined, Component \_ Rno, Qty \_ of \_ component)

   resource \_ ev \_ hierarchy(PI, Pname, Cseq, Cname)
   - expected \_ value(Rno, Expected \_ life \_ in \_ months, Expected \_ pattern \_ of \_ benefits, Expected \_ residual \_ value)
   - resource \_ price \_ level \_ series \_ mapping(Rtype, Series)

   eh(Eno, Date, Etype, Doc \_ type, Doc \_ no, Agent \_ id, Narrative)
   - rc(Eno, Line \_ no, AOacno, Qty, Value)

   agent(Agent \_ id, Agent \_ name, Address)

   share \_ stock(Rno, Nominal \_ value, Date \_ incorporated)
   - production \_ step \_ input \_ AOac(AOac \_ no, Job \_ no, Step \_ no)
   - production \_ step \_ output \_ AOac(AOac \_ no, Job \_ no, Step \_ no)

3. **Interpreter** (defined this chapter)
   - map \_ AOac \_ to \_ GLac(AO \_ ACno, GL \_ ACno)
   - map \_ AC \_ concept \_ to \_ GLac(Concept, GL \_ ACno)
   - map \_ etype \_ to \_ eproc(Etype, Event \_ handler)

   production \_ step \_ GLacs(Job \_ no, Step \_ no, M \_ GLac, L \_ GLac, OH \_ Incurred \_ GLac, OH \_ Applied \_ GLac)
   - unit \_ standard \_ cost(Rno, Date \_ set, Standard \_ cost)
   - department \_ responsible \_ for \_ manufacture(Rno, Department)
   - departmental \_ OH \_ rate(Department, Date \_ set, Rate, Basis \_ of \_ variation)

   allocation \_ of \_ OH(From \_ GLac, To \_ GLac, Proportion)
   - wip \_ GLac \_ class(GL \_ ACno, WIP \_ class)
   - department \_ GL \_ variance \_ account(Department, Variance \_ ac \_ description)
   - department \_ OH \_ flexible \_ budget(Dept, Fixed \_ OH, Var \_ OH \_ coefficient, Basis \_ of \_ variation)

**Table 10.2 A Full List of Tables used in the Manufacturing Database**
Formula Accounting ledger system. The extra tables are required because the Interpreter program is being asked to perform tasks presently performed by human accountants. Human accountants have access to a very large database of facts about the world and rules about accounting (in their heads), but they are slow and expensive data processors.

How likely is it, one wonders, that the labour-intensive accounting systems of today will be replaced by systems designed along the lines proposed in this thesis? The next section offers some answers to that question.

10.5 Chapter Summary and Evaluation of the Proposed Architecture

In Chapter 6, on the basis of two design principles distilled from observations in the preceding chapters about (a) the desirable properties of future accounting systems and (b) present trends in computer-based accounting, an architecture for future computer-based accounting systems was proposed in Figure 6.3(c). That architecture introduced the notion of a Formula Accounting general ledger which recorded journal entries and ledger balances in terms of formulae, not constant money-values. It also proposed that Formula Accounting journal entries should be prepared automatically, by computer programs, based on data in the firm's many special-purpose computer-based transaction processing systems. Automatic preparation of journal entries would, it was argued, enable a firm to run a Multiview Accounting system, where a number of Formula Accounting systems were maintained in parallel.
The purpose of this chapter was to investigate the problems associated with automatic generation of Formula Accounting journal entries. The methodology chosen for that investigation was to try to build the program shown in the lower path in Figure 10.1, i.e., to build a prototype Interpreter. Restricting attention to the lower path in Figure 10.1 is not a limitation, because the data requirements, accounting rules, and FA journal entries generated by the system are the same, in either the upper or lower path.

The investigations reported in this chapter indicate that once the data are available in an organized format, e.g., in REE format, it is possible to generate Formula Accounting journal entries relatively easily. Using little more than 2,000 lines of Prolog it was possible to define a computer program that generated correct HC, CPP, and CC/CPP accounting journal entries for many common types of economic events. In both the pseudo-code Interpreter in Figure 10.2, and the more general Case-structured Interpreter outlined in Figure 10.4, the key piece of information used to guide journal-entry generation is the Exchange Event's Event Type. Provided a human being (either directly, or indirectly through another computer program) has correctly specified an event as a "revenue recognition event", "capital raising event", "manufacturing event", or whatever, the rules of double-entry accounting are sufficiently regular for almost any sort of resource exchange to be modelled correctly and relatively easily.

The experience in building the prototype Interpreter also indicates that once the basic framework of core modules is in place, it is also relatively easy to "teach"
the Interpreter to handle new Event Types, and to use alternatives to Historical Cost valuation rules.

Turning to the future, it seems reasonable to predict that many future computer-based ledger systems will use Formula Accounting. Just as movie cameras add an extra dimension to still photography, so Formula Accounting adds an extra dimension to accounting. Its ability to model ledger balances as they change over time must make Formula Accounting attractive to many users of accounting ledger systems. Moreover, since Formula Accounting is relatively easy to implement (the prototype system required less than 1,000 lines of Prolog), it seems likely that many future ledger systems will offer Formula Accounting capabilities.

It also seems likely that users of future systems will want to generate their Formula Accounting journal entries automatically from data provided by their special-purpose sub-systems. This is, after all, what happens in an ad hoc fashion at present. However, whether future systems will follow Figure 10.1's upper or lower paths (or a combination) for generating FA journal entries is difficult to predict. Given the likely need for different REE data models and different sets of accounting rules for different businesses (certainly the databases and accounting rules differ across industries) it may well be easier to have the sub-systems generate the FA journal entries themselves.

If the upper path in Figure 10.1 is followed, where FA journal entries are prepared by the sub-systems themselves, the value of the last two chapters has
been to show that the data requirements and the rules of accounting are not too onerous.

On the other hand, if the lower path is followed, e.g., because one wants to certify the "quality" of the journal entries coming from a multiplicity of special-purpose systems, the work reported in this and the last chapter also indicates that it is possible to generate one or more accounting views of most (if not all) of a firm's transactions automatically. Experience in building the prototype Interpreter also indicates that in a lower-path architecture, definition and maintenance of an REE-like data model is likely to be a more difficult task than encoding the relevant accounting rules into an Interpreter.

An accountant using an REE-model-Interpreter approach sometimes has to choose between a "unconventional" representation of an Exchange Event and codification of a new event handler in the Interpreter. This choice was illustrated in this thesis by Exchange Event number 6, where declaration of a dividend was treated, possibly "unconventionally", as a reduction in firm's obligation to stock holders to pay a stream of dividends. The guideline proposed in this chapter was that it is better to keep the data model as simple and "conventional" as possible, and to add extra event handlers to the Interpreter for special cases. "Conventional" treatment of Exchange Events could be defined as part of Generally Accepted Accounting Principles.

(9,100 words)

---

14 Exchange-Event modelling forces the accountant to ask the question: What has been exchanged for what? In the case of dividend payments, a firm would not pay dividends if it was under no obligation to do so. An obligation clearly exists. However it is "unconventional" to model that obligation in the accounts.
Exhibit 10.1: HC and CC/CPP Output for the Exchange Events in Tables 9.1 and 9.2

This 50 page Exhibit shows output from four separate runs of the prototype Interpreter. Runs 1 and 2 show HC and CC/CPP (FIFO) interpretations of the Trading data in Chapter 9, Table 9.1. Database tables used in Runs 1 and 2 are reproduced in Appendix 5, titled "Trading Database". Runs 3 and 4 show HC and CC/CPP (Standard Cost) interpretations of the Manufacturing data in Chapter 9, Table 9.2. Database tables used in Runs 3 and 4 are reproduced in Appendix 6, titled "Manufacturing Database". The conceptual schemas for both databases are the same, and the contents of some tables are very similar. However, the Exchange Events (lines 636-662 in Trading_database and 582-635 in Manuf_database) are different.

Most of the following 49 pages were printed by procedures in the Formula Accounting system. The journal entries were generated by the Interpreter, from data in the respective databases. Comments have been added in bold-type boxes to guide the reader. Comments in Runs 2 through 4 assume the reader has read the comments and footnotes in Run 1. Footnotes throughout the Exhibit guide the reader to specific points of Prolog code in Appendices 1 through 4.

Standard output for each run of the Interpreter is:

(a) Diagnostic output that shows the Interpreter's progress through the Exchange Events;
(b) A Trial Balance at Report_date;
(c) A "Money Trial Balance" at Report_date;
(d) A list of journal entries, with formulae evaluated at Report_date.

Whilst the objective of the system is to maintain a set of ledger account balances (which is why they have been listed before the journal entries), it is probably easier to understand the output for each Run if one jumps first to the journal entries, e.g., pages 8-10 of Run 1.

Prolog code for the entire system is reproduced in Appendices 1 through 6. A summary of the content of those Appendices is given in Table 10.1 in this chapter. Further details are given on the front page of each Appendix. The one-page explanation at the start of Appendix 2 steps the reader through the code to the point where it actually writes out such gems as:

uncondit_formula(119, -5000, []).

(meaning Cr $5,000). It is probably worth reading. It should also remembered that the representation of formulae, like that on the line above, is intended to be simple enough for a human-being to read, yet complex enough to represent formulae to enable reasonable testing of the prototype Formula Accounting/Interpreter/Multiview Accounting system. The "reports" that follow are not intended as examples of reports from a Formula Accounting system.
RUN 1, CC/CPP, FIFO, Table 9.1 data

Comment: The command below, mv(900101,910101), invokes the Interpreter to process the Exchange Events in Table 9.1 from January 1, 1990 to Report Date, December 31, 1990, in batch mode. Output for Historical Cost accounting, FIFO inventory valuation, is presented in the next twelve pages, pp.2-13. (Output for a second run of the Interpreter for CC/CPP accounting, FIFO inventory valuation, is presented in pages 13 through 26.)

Beginning this page is diagnostic output for the HC, FIFO run. Initially, Ijiri’s causal double-entry is used to value each Event’s resource changes. The values so determined are listed as "ed(...)" predicates on the next page. (Only ed(3,2,...) was not valued. The value for this sale of inventory is determined later by the Interpreter.)

mv(900101,910101).
Beginning initialization

closing_dates(900101, 910101).

Now using money values exchanged to value non-monetary assets and obligations except for the following values accepted from input:
Calculating V for resource: 1403
Eno 2 Credit Purchase Lno 1 V= 7000
Eno 2 Credit Purchase Lno 2 V= 2000
Calculating V for resource: 1108
Calculating V for resource: 1404
Calculating V for resource: 1505
Generation of exchange money values complete.

ed( 1, 1, 11103, 100000, 100000).
ed( 1, 2, 31102,-30000,-75000).
ed( 1, 3, 31103,-10000,-25000).
ed( 2, 1, 11413, 1000, 7000).

1 Prolog code for this output commences at Interpreter, line 5, i.e., at line 5 of the Appendix titled "Interpreter". (The Exchange Events are coded in Trading_database, lines 616-642.) Line 7 of the Interpreter invokes line 18. Line 19 invokes the initialize_model procedure at lines 88-144. Within the initialization, line 89 wrote out this line, and line 100 invoked the procedure generate_eds in Interpreter, lines 293-397.

2 Within generate_eds, the main loop is lines 298 through 303. Line 298 locates the eh(…) predicates in Trading_database, lines 616-642, and line 312 locates the rc(…) predicates for each event header. Line 363 wrote out the "Calculating V" message above. At the end of procedure generate_eds, line 305 writes out this line, and line 306 in Interpreter directs the system to list all the ed(…) predicates.

Exhibit 10.1, page 2
Comment: Below, the Interpreter prompts for the Accounting rules it is to follow in this run. "hc." and "f." are entered to select HC and FIFO.

Here are the present values of the Multiview Interpreter's switches

price_level_accounting_switch(HistoricalCost).

fifo_lifo_switch(standard_cost).

Change switches? Enter y. or n. (in lower case followed by a dot):
This Interpreter handles Historical Cost (hc),
CPP (cpp) or CC/CPP (cc/cpp) accounting.

Enter choice, either hc., cpp. or cc/cpp. (in lower case followed by a dot): hc.
This Interpreter handles either fifo (f),
lifo (l) or std. cost (s) inventory valuation.

Enter choice, either f., l. or s. (in lower case followed by a dot): f.
Comment: This page shows diagnostic output as the Interpreter works its way through the eight events from Table 9.1. The inventory_value records displayed during processing of Event 2 show unit costs for the Widgets 1 and 2 were $7 and $2, respectively. As the start and end times on this page show, Prolog is quite slow: generating and posting the simple HC journal entries shown on the next 5 pages took 9 seconds, elapsed time.

Here are the values of the Multiview Interpreter's switches for this run for accounting period from 900101 to Report date 910101:

Run date and time started: "1991-01-29" "22:33:32.877" ← start

price_level_accounting_switch(HistoricalCost).

diff lifo_switch(fifo).

----- Period starting 900101 and finishing 910101 -----
CALLING issue_shares for event 1 of type Issue Shares
issue_shares called for event 1
CALLING simple for event 2 of type Credit Purchase
simple called for event 2
inventory_value(1032, 2, 900201, 7, 1000)
inventory_value(1033, 2, 900201, 2, 1000)
CALLING credit_sale for event 3 of type Credit Sale
credit_sale called for event 3
CALLING simple for event 4 of type Credit Purchase
simple called for event 4
Starting straight line depreciation for Rno= 1108
CALLING cash_payment for event 5 of type Cash Payment
cash_payment called for event 5
CALLING simple for event 6 of type Declare Dividend
simple called for event 6
CALLING simple for event 7 of type Wages Payment
simple called for event 7
Zero life shrinking value asset, expensing immediately
CALLING donation for event 8 of type Donation
donation called for event 8

ALL EVENTS UP TO REPORT DATE NOW PROCESSED
Run date and time completed: "1991-01-29" "22:33:41.202" ← end

Exhibit 10.1, page 4
Comment: Trial Balance for Historical Cost accounting, FIFO inventory flow assumption, Table 9.1 data. This "report" shows the ledger account formulae for all accounts with balances. For human consumption, formulae are evaluated, as at January 1, 1991, on the lines commencing "***" (for each formula) and "Overall" (an account balance may depend on many formulae).

Trial Balance\(^3\) for REEFA LTD, date 910101, for Historical Cost, fifo

**GL_AC = 11103 <Bank 1**
- uncondit_formula(105, 95400,[])
  - *** On date 910101 formula evaluates to 95400 ***
  - Overall account balance on date 910101 is 95400

**GL_AC = 11208 <Customer 1**
- uncondit_formula(118, 5000,[])
  - *** On date 910101 formula evaluates to 5000 ***
  - Overall account balance on date 910101 is 5000

**GL_AC = 11413 <Widget 1**
- uncondit_formula(111, 3500,[])
  - *** On date 910101 formula evaluates to 3500 ***
  - Overall account balance on date 910101 is 3500

**GL_AC = 11414 <Widget 2**
- uncondit_formula(112, 2000,[])
  - *** On date 910101 formula evaluates to 2000 ***
  - Overall account balance on date 910101 is 2000

**GL_AC = 12801 <1990 Ford Delivery Truck**
- uncondit_formula(131, 36500,[])
  - *** On date 910101 formula evaluates to 36500 ***
  - Overall account balance on date 910101 is 36500

**GL_AC = 12901 <Accum Depr - 1990 Ford Truck**
- formula(127, 137,<, 138, 136, 139)
  - uncondit_formula(137,-32953,[])
  - uncondit_formula(137, 1,[d, 1])
  - uncondit_formula(138, 3652.5,[])
  - uncondit_formula(136, 329304.449007528,[])
  - uncondit_formula(136, -9.99315557363215,[d, 1])
  - uncondit_formula(139,-36500,[])
  - *** On date 910101 formula evaluates to -2838.05612594076 ***
  - Overall account balance on date 910101 is -2838.05612594076

**GL_AC = 21107 <Supplier 1**
- uncondit_formula(113,-5000,[])
  - *** On date 910101 formula evaluates to -5000 ***
  - Overall account balance on date 910101 is -5000

**GL_AC = 21108 <Supplier 2**
- uncondit_formula(140,-36500,[])
  - *** On date 910101 formula evaluates to -36500 ***
  - Overall account balance on date 910101 is -36500

---

\(^3\) Prolog source code to generate the Trial Balance is in Formula, lines 197-245.

Exhibit 10.1, page 5
GL_AC = 21502 <Dividend Payable - SH1 >,
  uncondit_formula(148,-600,[I])
  *** On date 910101 formula evaluates to -600 ***
  Overall account balance on date 910101 is -600

GL_AC = 21503 <Dividend Payable - SH2 >,
  uncondit_formula(149,-200,[I])
  *** On date 910101 formula evaluates to -200 ***
  Overall account balance on date 910101 is -200

GL_AC = 31101 <Issued Capital >,
  uncondit_formula(106,-80000,[I])
  *** On date 910101 formula evaluates to -80000 ***
  Overall account balance on date 910101 is -80000

GL_AC = 31102 <Share Premium Reserve >,
  uncondit_formula(107,-20000,[I])
  *** On date 910101 formula evaluates to -20000 ***
  Overall account balance on date 910101 is -20000

GL_AC = 31800 <Retained Earnings >,
  uncondit_formula(147,800,[I])
  *** On date 910101 formula evaluates to 800 ***
  Overall account balance on date 910101 is 800

GL_AC = 32100 <Sales Revenue >,
  uncondit_formula(119,-5000,[I])
  *** On date 910101 formula evaluates to -5000 ***
  Overall account balance on date 910101 is -5000

GL_AC = 33100 <Cost of Sales >,
  uncondit_formula(120,3500,[I])
  *** On date 910101 formula evaluates to 3500 ***
  Overall account balance on date 910101 is 3500

GL_AC = 33401 <Office Wages Expense >,
  uncondit_formula(155,500,[I])
  *** On date 910101 formula evaluates to 500 ***
  Overall account balance on date 910101 is 500

GL_AC = 33407 <Depreciation Expense - 1990 Ford Truck >,
  formula(126,133,<,134,132,135)
  uncondit_formula(133,-32963,[I])
  uncondit_formula(133,1,[d,1])
  uncondit_formula(134,3652.5,[I])
  uncondit_formula(132,-329304.449007528,[I])
  uncondit_formula(132,9.99315537303215,[d,1])
  uncondit_formula(135,36500,[I])
  *** On date 910101 formula evaluates to 2838.05612594076 ***
  Overall account balance on date 910101 is 2838.05612594076

GL_AC = 33408 <Donation Expense >,
  uncondit_formula(158,100,[I])
  *** On date 910101 formula evaluates to 100 ***
  Overall account balance on date 910101 is 100

Overall ledger balance on date 910101 is 0

Exhibit 10.1, page 6
Comment: This report, termed a "Money Trial Balance", prints money values only for balances, not formulae. It shows Historical cost accounting, FIFO inventory flow, Table 9.1 data.

This report adds nothing to the information provided by the previous report. Its purpose is mainly to reassure the reader that the detail of a FA system can be hidden from the user.

MONEY TRIAL BALANCE, REEFA LTD, as at 910101

for Historical Cost, fifo

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance (cents)</th>
<th>CR Balance (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11103</td>
<td>Bank 1</td>
<td>9540000</td>
<td></td>
</tr>
<tr>
<td>11208</td>
<td>Customer 1</td>
<td>500000</td>
<td></td>
</tr>
<tr>
<td>11413</td>
<td>Widget 1</td>
<td>350000</td>
<td></td>
</tr>
<tr>
<td>11414</td>
<td>Widget 2</td>
<td>200000</td>
<td></td>
</tr>
<tr>
<td>12801</td>
<td>1990 Ford Delivery Truck</td>
<td></td>
<td>3650000</td>
</tr>
<tr>
<td>12901</td>
<td>Accum Depr - 1990 Ford Truck</td>
<td></td>
<td>-283806</td>
</tr>
<tr>
<td>21107</td>
<td>Supplier 1</td>
<td></td>
<td>-500000</td>
</tr>
<tr>
<td>21108</td>
<td>Supplier 2</td>
<td></td>
<td>-3650000</td>
</tr>
<tr>
<td>21502</td>
<td>Dividend Payable - SH1</td>
<td></td>
<td>-60000</td>
</tr>
<tr>
<td>21503</td>
<td>Dividend Payable - SH2</td>
<td></td>
<td>-20000</td>
</tr>
<tr>
<td>31101</td>
<td>Issued Capital</td>
<td></td>
<td>-8000000</td>
</tr>
<tr>
<td>31102</td>
<td>Share Premium Reserve</td>
<td></td>
<td>-2000000</td>
</tr>
<tr>
<td>31800</td>
<td>Retained Earnings</td>
<td>80000</td>
<td>-500000</td>
</tr>
<tr>
<td>32100</td>
<td>Sales Revenue</td>
<td>350000</td>
<td></td>
</tr>
<tr>
<td>33100</td>
<td>Cost of Sales</td>
<td>50000</td>
<td></td>
</tr>
<tr>
<td>33401</td>
<td>Office Wages Expense</td>
<td>283806</td>
<td></td>
</tr>
<tr>
<td>33407</td>
<td>Depreciation Expense - 1990 Ford Truck</td>
<td>10000</td>
<td></td>
</tr>
<tr>
<td>33408</td>
<td>Donation Expense</td>
<td>15013806</td>
<td>-15013806</td>
</tr>
</tbody>
</table>

---

Prolog code to generate the report is Formula, lines 246-308. The rounding-to-cents routine is in Library, lines 87-109.

Exhibit 10.1, page 7
Comment: Listing of Formula Accounting journal entries generated by the Interpreter for HC, FIFO accounting rules, and Exchange Events in Table 9.1. This listing takes three pages.

Journal entries for REEFA LTD, for period from 900101 to Report date 910101 for Historical Cost, fifo

jeh(0, 900101, 1, Issue of 40000 $2 shares at a premium)
   jed(0, 1, 11103 <Bank 1 >, 100).
   uncondit_formula(100, 100000,[])
   *** On date 910101 formula evaluates to 100000 ***
   jed(0, 2, 31101 <Issued Capital >, 101).
   uncondit_formula(101, -60000,[])
   *** On date 910101 formula evaluates to -60000 ***
   jed(0, 3, 31102 <Share Premium Reserve >, 102).
   uncondit_formula(102, -15000,[])
   *** On date 910101 formula evaluates to -15000 ***
   jed(0, 4, 31101 <Issued Capital >, 103).
   uncondit_formula(103, -20000,[])
   *** On date 910101 formula evaluates to -20000 ***
   jed(0, 5, 31102 <Share Premium Reserve >, 104).
   uncondit_formula(104, -5000,[])
   *** On date 910101 formula evaluates to -5000 ***

jeh(1, 900201, 2, Purchase of 1000 ea. of Widgets 1 and 2 from Supplier 1)
   jed(1, 1, 11413 <Widget 1 >, 108).
   uncondit_formula(108, 7000,[])
   *** On date 910101 formula evaluates to 7000 ***
   jed(1, 2, 11414 <Widget 2 >, 109).
   uncondit_formula(109, 2000,[])
   *** On date 910101 formula evaluates to 2000 ***
   jed(1, 3, 21107 <Supplier 1 >, 110).
   uncondit_formula(110, -9000,[])
   *** On date 910101 formula evaluates to -9000 ***

jeh(2, 900301, 3, Sale of 500 Widget 1's to Customer 1)
   jed(2, 1, 11208 <Customer 1 >, 114).
   uncondit_formula(114, 5000,[])
   *** On date 910101 formula evaluates to 5000 ***
   jed(2, 2, 32100 <Sales Revenue >, 115).
   uncondit_formula(115, -5000,[])
   *** On date 910101 formula evaluates to -5000 ***
   jed(2, 3, 33100 <Cost of Sales >, 116).
   uncondit_formula(116, 3500,[])
   *** On date 910101 formula evaluates to 3500 ***

5 Prolog code to list journal entries is shown in Formula, lines 2-192.

Exhibit 10.1, page 8
jed(2, 4, 11413 <Widget 1>, 117).
  uncondit_formula(117, -3500, []),
  *** On date 910101 formula evaluates to -3500 ***

jeh(3, 900323, 4, Purchase of 1990 Ford Delivery Truck from Supplier 2)
  jed(3, 1, 12801 <1990 Ford Delivery Truck>, 121).
  uncondit_formula(121, 36500, []),
  *** On date 910101 formula evaluates to 36500 ***

  jed(3, 2, 33407 <Depreciation Expense - 1990 Ford Truck>, 122).
  formula(122, 122, <, 123, 124, 125)
  uncondit_formula(122, -32953, []),
  uncondit_formula(122, 1, [[d, 1]]),
  uncondit_formula(123, 3652.5, []),
  uncondit_formula(124, -329304.449007528, []),
  uncondit_formula(124, -9.99315537303215, [[d, 1]]),
  uncondit_formula(125, 36500, []),
  *** On date 910101 formula evaluates to 2838.05612594076 ***

  jed(3, 3, 12901 <Accum Depr - 1990 Ford Truck>, 123).
  formula(123, 126, <, 127, 128, 129)
  uncondit_formula(126, -32953, []),
  uncondit_formula(126, 1, [[d, 1]]),
  uncondit_formula(127, 3652.5, []),
  uncondit_formula(128, -329304.449007528, []),
  uncondit_formula(128, -9.99315537303215, [[d, 1]]),
  uncondit_formula(129, -36500, []),
  *** On date 910101 formula evaluates to -2838.05612594076 ***

  jed(3, 4, 21108 <Supplier 2>, 124).
  uncondit_formula(130, -36500, []),
  *** On date 910101 formula evaluates to -36500 ***

jeh(4, 900401, 5, Pay Supplier 1 $4,000)
  jed(4, 1, 21107 <Supplier 1>, 129).
  uncondit_formula(141, 4000, []),
  *** On date 910101 formula evaluates to 4000 ***

  jed(4, 2, 11103 <Bank 1>, 130).
  uncondit_formula(142, -4000, []),
  *** On date 910101 formula evaluates to -4000 ***

jeh(5, 900501, 6, Declaration of dividend of $0.02 per share)
  jed(5, 1, 31800 <Retained Earnings>, 131).
  uncondit_formula(143, 600, []),
  *** On date 910101 formula evaluates to 600 ***

  jed(5, 2, 31800 <Retained Earnings>, 132).
  uncondit_formula(144, 200, []),
  *** On date 910101 formula evaluates to 200 ***

  jed(5, 3, 21502 <Dividend Payable - SH1>, 133).
  uncondit_formula(145, -600, []),
  *** On date 910101 formula evaluates to -600 ***

  jed(5, 4, 21503 <Dividend Payable - SH2>, 134).

Exhibit 10.1, page 9
uncondit_formula(146,-200,[])  
*** On date 910101 formula evaluates to -200 ***

jeh(6, 900601, 7, Pay office wages of $500 for work in May)  
jed(6, 1, 11800 <Prepaid Wages >, 138).  
uncondit_formula(150, 500,[])  
*** On date 910101 formula evaluates to 500 ***

jed(6, 2, 33401 <Office Wages Expense >, 139).  
uncondit_formula(151, 500,[])  
*** On date 910101 formula evaluates to 500 ***

jed(6, 3, 11800 <Prepaid Wages >, 140).  
uncondit_formula(152,-500,[])  
*** On date 910101 formula evaluates to -500 ***

jed(6, 4, 11103 <Bank 1 >, 141).  
uncondit_formula(153,-500,[])  
*** On date 910101 formula evaluates to -500 ***

jeh(7, 900701, 8, Donation to Red Cross)  
jed(7, 1, 11103 <Bank 1 >, 144).  
uncondit_formula(156,-100,[])  
*** On date 910101 formula evaluates to -100 ***

jed(7, 2, 33408 <Donation Expense >, 145).  
uncondit_formula(157, 100,[])  
*** On date 910101 formula evaluates to 100 ***

yes

---

**Comment:** The commands in bold type below first "close" the revenue and expense accounts to an account called Profit and Loss (i.e., evaluate balance formulae, then post those figures to Profit and Loss), then list the new journal entry created by close_revenue_acs(with formulae evaluated on January 1, 1991).

close_revenue_acs(910101).  
Entering close_revenue_acs6 for period ending 910101
yes
list_j(e(8,910101).

jeh(8, 910101, closing_acs, Closing revenue and expense for period)  
jed(8, 1, 31900 <Profit and Loss >, 147).  
uncondit_formula(159,-5000,[])  
*** On date 910101 formula evaluates to -5000 ***

jed(8, 2, 32100 <Sales Revenue >, 148).  
uncondit_formula(160, 5000,[])  
*** On date 910101 formula evaluates to 5000 ***

jed(8, 3, 31900 <Profit and Loss >, 149).

---

6 Prolog code to close_revenue_acs is in Formula, lines 660-711.  
Exhibit 10.1, page 10
Comment: The tb command in bold type below generates a new Trial Balance showing ledger accounts with balances to be carried forward to the next accounting period, and with formulae evaluated on January 1, 1991. This "report" is still for HC, FIFO accounting rules, Table 9.1 data.

\[ \text{tb(910101).} \]

Trial Balance for REEFA LTD, date 910101, for Historical Cost, fifo

\[
\begin{align*}
\text{GL\_AC} &= 11103 <\text{Bank 1} >, \\
& \quad \text{uncondit\_formula(105, 95400,[])} \\
& \quad \text{*** On date 910101 formula evaluates to 95400 ***} \\
& \text{Overall account balance on date 910101 is 95400} \\
\text{GL\_AC} &= 11208 <\text{Customer 1} >, \\
& \quad \text{uncondit\_formula(118, 5000,[])} \\
& \quad \text{*** On date 910101 formula evaluates to 5000 ***} \\
& \text{Overall account balance on date 910101 is 5000} \\
\text{GL\_AC} &= 11413 <\text{Widget 1} >, \\
& \quad \text{uncondit\_formula(111, 3500,[])} \\
& \quad \text{*** On date 910101 formula evaluates to 3500 ***} \\
& \text{Overall account balance on date 910101 is 3500} \\
\text{GL\_AC} &= 11414 <\text{Widget 2} >, \\
& \quad \text{uncondit\_formula(112, 2000,[])}
\end{align*}
\]

Exhibit 10.1, page 11
*** On date 910101 formula evaluates to 2000 ***
Overall account balance on date 910101 is 2000

GL_AC = 12801 <1990 Ford Delivery Truck >,
  uncondit_formula(131, 36500,[])
*** On date 910101 formula evaluates to 36500 ***
Overall account balance on date 910101 is 36500

GL_AC = 12901 <Accum Depr - 1990 Ford Truck >,
  formula(127, 137,<, 138, 136, 139)
  uncondit_formula(137,-32953,[])
  uncondit_formula(138, 3652.5,[])
  uncondit_formula(136, 329304.449007528,[])
  uncondit_formula(136, -9.99315537303215,[])
  uncondit_formula(139,-36500,[])
*** On date 910101 formula evaluates to -2838.05612594076 ***
Overall account balance on date 910101 is -2838.05612594076

GL_AC = 21107 <Supplier 1 >,
  uncondit_formula(113,-5000,[])
*** On date 910101 formula evaluates to -5000 ***
Overall account balance on date 910101 is -5000

GL_AC = 21108 <Supplier 2 >,
  uncondit_formula(140,-36500,[])
*** On date 910101 formula evaluates to -36500 ***
Overall account balance on date 910101 is -36500

GL_AC = 21502 <Dividend Payable - SH1 >,
  uncondit_formula(148,-600,[])
*** On date 910101 formula evaluates to -600 ***
Overall account balance on date 910101 is -600

GL_AC = 21503 <Dividend Payable - SH2 >,
  uncondit_formula(149,-200,[])
*** On date 910101 formula evaluates to -200 ***
Overall account balance on date 910101 is -200

GL_AC = 31101 <Issued Capital >,
  uncondit_formula(106,-80000,[])
*** On date 910101 formula evaluates to -80000 ***
Overall account balance on date 910101 is -80000

GL_AC = 31102 <Share Premium Reserve >,
  uncondit_formula(107,-20000,[])
*** On date 910101 formula evaluates to -20000 ***
Overall account balance on date 910101 is -20000

GL_AC = 31800 <Retained Earnings >,
  uncondit_formula(147,800,[])
*** On date 910101 formula evaluates to 800 ***
Overall account balance on date 910101 is 800

GL_AC = 31900 <Profit and Loss >,
  uncondit_formula(169,1938.05612594076,[])
*** On date 910101 formula evaluates to 1938.05612594076 ***
Overall account balance on date 910101 is 1938.05612594076

GL_AC = 33407 <Depreciation Expense - 1990 Ford Truck >,

Exhibit 10.1, page 12
Overall account balance on date 910101 is 0

---

Comment: A Money Trial Balance version of the above follows, i.e., Historical cost accounting, FIFO inventory flow, Table 9.1 data. Compared to the report on page 7, the revenue and expense accounts now evaluate to zero balances and are not listed.

money_tb(910101).
MONEY TRIAL BALANCE, REEFA LTD, as at 910101 for Historical Cost, fifo

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance (cents)</th>
<th>CR Balance (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11103</td>
<td>Bank 1</td>
<td>9540000</td>
<td></td>
</tr>
<tr>
<td>11208</td>
<td>Customer 1</td>
<td>500000</td>
<td></td>
</tr>
<tr>
<td>11413</td>
<td>Widget 1</td>
<td>350000</td>
<td></td>
</tr>
<tr>
<td>11414</td>
<td>Widget 2</td>
<td>200000</td>
<td></td>
</tr>
<tr>
<td>12801</td>
<td>1990 Ford Delivery Truck</td>
<td></td>
<td>3650000</td>
</tr>
<tr>
<td>12901</td>
<td>Accum Depr - 1990 Ford Truck</td>
<td></td>
<td>-283806</td>
</tr>
<tr>
<td>21107</td>
<td>Supplier 1</td>
<td></td>
<td>-500000</td>
</tr>
<tr>
<td>21108</td>
<td>Supplier 2</td>
<td></td>
<td>-3650000</td>
</tr>
<tr>
<td>21502</td>
<td>Dividend Payable - SH1</td>
<td></td>
<td>-60000</td>
</tr>
<tr>
<td>21503</td>
<td>Dividend Payable - SH2</td>
<td></td>
<td>-20000</td>
</tr>
<tr>
<td>31101</td>
<td>Issued Capital</td>
<td></td>
<td>-8000000</td>
</tr>
<tr>
<td>31102</td>
<td>Share Premium Reserve</td>
<td></td>
<td>-2000000</td>
</tr>
<tr>
<td>31800</td>
<td>Retained Earnings</td>
<td>80000</td>
<td></td>
</tr>
<tr>
<td>31900</td>
<td>Profit and Loss</td>
<td>193806</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>14513806</td>
<td>-14513806</td>
</tr>
</tbody>
</table>

---

Yes
Comment: The command below, mv(900101,910101), re-runs the Interpreter to process again all Exchange Events from Table 9.1 from January 1, 1990 through December 31, 1990, in batch mode, this time for CC/CPP accounting, not historical cost.

As in the HC run, the order of presentation is (a) Diagnostic Output, (b) Trial Balance, (c) Money Trial Balance, (d) Listing of journal entries. This is followed by a command to close_revenue_acs and a new Trial Balance and Money Trial Balance.

Diagnostic Output commences on this page. The Interpreter has not repeated causal double-entry reasoning, since the facts are unchanged from Run 1. CC/CPP, FIFO accounting rules are selected in the lower half of this page. The inventory_value predicates shown 3 and 4 lines below this box are leftovers from the last run. They show stocks of resources Widget 1 and 2 just before they were purged for this new run.

mv(900101,910101).
Beginning initialization

closing_dates(900101, 910101).

inventory_value(1032, 2, 900201, 7, 500).
inventory_value(1033, 2, 900201, 2, 1000).

Here are the present values of the Multiview Interpreter's switches

price_level_accounting_switch(HistoricalCost).

fifo_lifo_switch(fifo).

Change switches? Enter y. or n. (in lower case followed by a dot):
This Interpreter handles Historical Cost (hc),
CPP (cpp) or CC/CPP (cc/cpp) accounting.
Enter choice, either hc., cpp. or cc/cpp. (in lower case followed by a dot): cc/cpp.
This Interpreter handles either fifo (f),
lifo (l) or std. cost (s) inventory valuation.
Enter choice, either f., l. or s. (in lower case followed by a dot): f.
Comment: This page shows diagnostic output as the Interpreter works its way through the eight Exchange Events. The "Extrapolating back" message during processing of Event 1 is because the date of incorporation was defined as November 1, 1989, and the price_index predicate was defined (Trading database, lines 858-9) for a later period, from January 1, 1990 for four years. Linear interpolation was used for index values at other dates. Generating and posting the CC/CPP entries took much longer than for HC, principally because of (i) all the indexing calculations involved, (ii) the extra journal entry lines to be written, and (iii) the extra time to post the extra journal entries. As shown by the run start and end times on this page, the FA journal entries shown on the next 7 pages took 57 seconds, elapsed time, to generate.

Here are the values of the Multiview Interpreter's switches for this run:

Run date and time started: "1991-01-29" "22:34:12.999"

price_level_accounting_switch(CurrentCost Accounting).

fifo_lifo_switch(fifo).

---- Period starting 900101 and finishing 910101 -----
CALLING issue_shares for event 1 of type Issue Shares
issue_shares called for event 1
Extrapolating back 61 days for time series. (Suspect for long times.)
CALLING simple for event 2 of type Credit Purchase
simple called for event 2
inventory_value(1032, 2, 900201, 7, 1000)
inventory_value(1033, 2, 900201, 2, 1000)
CALLING credit_sale for event 3 of type Credit Sale
credit_sale called for event 3
CALLING simple for event 4 of type Credit Purchase
simple called for event 4
Starting straight line depreciation for Rno= 1108
CALLING cash_payment for event 5 of type Cash Payment
cash_payment called for event 5
CALLING simple for event 6 of type Declare Dividend
simple called for event 6
CALLING simple for event 7 of type Wages Payment
simple called for event 7
Zero life shrinking value asset, expensing immediately
CALLING donation for event 8 of type Donation
donation called for event 8

ALL EVENTS UP TO REPORT DATE NOW PROCESSED
Run date and time completed: "1991-01-29" "22:35:09.917"
Comment: Trial Balance for CC/CPP accounting, FIFO inventory flow assumption, Table 9.1 data.

Trial Balance for REEFA LTD, date 910101, for Current Cost Accounting, fifo

GL_AC = 11103 <Bank 1>
  uncondit_formula(106, 95400,[])
  *** On date 910101 formula evaluates to 95400 ***
  Overall account balance on date 910101 is 95400

GL_AC = 11208 <Customer 1>
  uncondit_formula(126, 5000,[])
  *** On date 910101 formula evaluates to 5000 ***
  Overall account balance on date 910101 is 5000

GL_AC = 11413 <Widget 1>
  uncondit_formula(116, 34.7054431926158,[[si, 1]])
  *** On date 910101 formula evaluates to 3817.36120537529 ***
  Overall account balance on date 910101 is 3817.36120537529

GL_AC = 11414 <Widget 2>
  uncondit_formula(118, 19.8316818243519,[[si, 1]])
  *** On date 910101 formula evaluates to 2181.34926021445 ***
  Overall account balance on date 910101 is 2181.34926021445

GL_AC = 12801 <1990 Ford Delivery Truck>
  uncondit_formula(141, 357.081157091203,[[sp, 1]])
  *** On date 910101 formula evaluates to -3053.94147207029 ***
  Overall account balance on date 910101 is -3053.94147207029

GL_AC = 12901 <Accum Depr - 1990 Ford Truck>
  uncondit_formula(147, -32953,[])
  uncondit_formula(148, 3652.5,11)
  uncondit_formula(146, 3221.60037498326,[[sp, 1]])
  *** On date 910101 formula evaluates to -3053.94147207029 ***
  Overall account balance on date 910101 is -3053.94147207029

GL_AC = 21107 <Supplier 1>
  uncondit_formula(119,-5000,[])
  *** On date 910101 formula evaluates to -5000 ***
  Overall account balance on date 910101 is -5000

GL_AC = 21108 <Supplier 2>
  uncondit_formula(150,-36500,[])
  *** On date 910101 formula evaluates to -36500 ***
  Overall account balance on date 910101 is -36500

GL_AC = 21502 <Dividend Payable - SH1>
  uncondit_formula(162,-600,[])
  *** On date 910101 formula evaluates to -600 ***
  Overall account balance on date 910101 is -600

GL_AC = 21503 <Dividend Payable - SH2>
Overall account balance on date 910101 is -200

GL_AC = 31101 <Issued Capital >,
uncondit_formula(108, -806.73660960795,[[g, 1]])
*** On date 910101 formula evaluates to -84704.5831032568 ***
Overall account balance on date 910101 is -84704.5831032568

GL_AC = 31102 <Share Premium Reserve >,
uncondit_formula(109, -193.263390392051,[[g, 1]])
*** On date 910101 formula evaluates to -20291.9945832583 ***
Overall account balance on date 910101 is -20291.9945832583

GL_AC = 31401 <Gearing Gain >,
uncondit_formula(107, 585.237294089555,[[g, 1]])
*** On date 910101 formula evaluates to 3347.91301391972 ***
Overall account balance on date 910101 is 3347.91301391972

GL_AC = 31501 <CCA Gain above General Prices >,
uncondit_formula(117, -54.5371250169674,[[si, 1]])
uncondit_formula(117, -357.081157091203,[[sp, 1]])
uncondit_formula(117, 415.633339065324,[[g, 1]])
uncondit_formula(117, 0,[[sw, 1]])
*** On date 910101 formula evaluates to -1635.1154840262 ***
Overall account balance on date 910101 is -1635.1154840262

GL_AC = 31800 <Retained Earnings >,
uncondit_formula(161, 7.887070707070706,[[g, 1]])
*** On date 910101 formula evaluates to 826.397306397295 ***
Overall account balance on date 910101 is 826.397306397295

GL_AC = 32100 <Sales Revenue >,
uncondit_formula(127, -49.5994024986421,[[g, 1]])
*** On date 910101 formula evaluates to -5207.76751765341 ***
Overall account balance on date 910101 is -5207.76751765341

GL_AC = 33100 <Cost of Sales >,
uncondit_formula(128, 34.9835014691453,[[g, 1]])
*** On date 910101 formula evaluates to 3673.1479275142 ***
Overall account balance on date 910101 is 3673.1479275142

GL_AC = 33401 <Office Wages Expense >,
uncondit_formula(172, 4.89873927038627,[[g, 1]])
*** On date 910101 formula evaluates to 514.350858369093 ***
Overall account balance on date 910101 is 514.350858369093

GL_AC = 33407 <Depreciation Expense - 1990 Ford Truck >,
formula(136, 143, <, 144, 142, 145)
uncondit_formula(143, -32953,[])
uncondit_formula(143, 1,[[d, 1]])
uncondit_formula(144, 3652.5,[])
uncondit_formula(142, -3221.60037498326,[[sp, 1]])
uncondit_formula(142, -0.097763492701219,[[d, 1],[sp, 1]])
uncondit_formula(145, 357.081157091203,[[sp, 1]])
*** On date 910101 formula evaluates to 3053.94147207029 ***
Overall account balance on date 910101 is 3053.94147207029

Exhibit 10.1, page 17
GL_AC = 33408 <Donation Expense

uncondit_formula(176, 0,[])
uncondit_formula(176, .97582153529256,[[g, 1]])

*** On date 910101 formula evaluates to 102.457921453379 ***
Overall account balance on date 910101 is 102.457921453379

Overall ledger balance on date 910101 is 0

-------------------------------------

Comment: A Money Trial Balance version of the above follows, i.e., CC/CPP accounting, FIFO inventory flow assumption, Table 9.1 data.

MONEY TRIAL BALANCE, REEFA LTD, as at 910101
for Current Cost Accounting, fifo

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance (cents)</th>
<th>CR Balance (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11103</td>
<td>Bank 1</td>
<td>9540000</td>
<td></td>
</tr>
<tr>
<td>11208</td>
<td>Customer 1</td>
<td>500000</td>
<td></td>
</tr>
<tr>
<td>11413</td>
<td>Widget 1</td>
<td>381736</td>
<td></td>
</tr>
<tr>
<td>11414</td>
<td>Widget 2</td>
<td>218135</td>
<td></td>
</tr>
<tr>
<td>12801</td>
<td>1990 Ford Delivery Truck</td>
<td>3927648</td>
<td></td>
</tr>
<tr>
<td>12901</td>
<td>Accum Depr - 1990 Ford Truck</td>
<td></td>
<td>-305394</td>
</tr>
<tr>
<td>21107</td>
<td>Supplier 1</td>
<td></td>
<td>-500000</td>
</tr>
<tr>
<td>21108</td>
<td>Supplier 2</td>
<td></td>
<td>-3650000</td>
</tr>
<tr>
<td>21502</td>
<td>Dividend Payable - SH1</td>
<td></td>
<td>-60000</td>
</tr>
<tr>
<td>21503</td>
<td>Dividend Payable - SH2</td>
<td></td>
<td>-20000</td>
</tr>
<tr>
<td>31101</td>
<td>Issued Capital</td>
<td></td>
<td>-8470458</td>
</tr>
<tr>
<td>31102</td>
<td>Share Premium Reserve</td>
<td></td>
<td>-2029199</td>
</tr>
<tr>
<td>31401</td>
<td>Gearing Gain</td>
<td>334791</td>
<td></td>
</tr>
<tr>
<td>31501</td>
<td>CCA Gain above General Prices</td>
<td></td>
<td>-163512</td>
</tr>
<tr>
<td>31800</td>
<td>Retained Earnings</td>
<td>82640</td>
<td></td>
</tr>
<tr>
<td>32100</td>
<td>Sales Revenue</td>
<td></td>
<td>-520777</td>
</tr>
<tr>
<td>33100</td>
<td>Cost of Sales</td>
<td>367315</td>
<td></td>
</tr>
<tr>
<td>33401</td>
<td>Office Wages Expense</td>
<td>51435</td>
<td></td>
</tr>
<tr>
<td>33407</td>
<td>Depreciation Expense - 1990 Ford Truck</td>
<td>305394</td>
<td></td>
</tr>
<tr>
<td>33408</td>
<td>Donation Expense</td>
<td>10246</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>15719340</td>
<td>-15719340</td>
</tr>
</tbody>
</table>

Exhibit 10.1, page 18
Comment: Journal entries for CC/CPP accounting, FFIO inventory flow assumption, Table 9.1 data.

Journal entries for REEFA LTD, for period from 900101 to Report date 910101 for Current Cost Accounting, fifo

jeh(0, 900101, 1, Issue of 40000 $2 shares at a premium)
jed(0, 1, 11103 <Bank 1 >, 100).
  uncondit_formula(100, 100000,[])
*** On date 910101 formula evaluates to 100000 ***
jed(0, 2, 31401 <Gearing Gain >, 101).
  uncondit_formula(101, 1000,[[g, 1]])
  uncondit_formula(101, -100000,[])
*** On date 910101 formula evaluates to 4996.57768651471 ***
jed(0, 3, 31101 <Issued Capital >, 102).
  uncondit_formula(102, -605.052457205962,[[g, 1]])
*** On date 910101 formula evaluates to -63528.4373274425 ***
jed(0, 4, 31102 <Share Premium Reserve >, 103).
  uncondit_formula(103, -144.947542794039,[[g, 1]])
*** On date 910101 formula evaluates to -15218.9959374437 ***
jed(0, 5, 31101 <Issued Capital >, 104).
  uncondit_formula(104, -201.684152401987,[[g, 1]])
*** On date 910101 formula evaluates to -21176.1457758142 ***
jed(0, 6, 31102 <Share Premium Reserve >, 105).
  uncondit_formula(105, -48.3158475980129,[[g, 1]])
*** On date 910101 formula evaluates to -5072.99864581457 ***
jeh(1, 900201, 2, Purchase of 1000 ea. of Widgets 1 and 2 from Supplier 1)
jed(1, 1, 11413 <Widget 1 >, 110).
  uncondit_formula(110, 69.4108863852316,[[si, 1]])
*** On date 910101 formula evaluates to 7634.72241075058 ***
jed(1, 2, 31501 <CCA Gain above General Prices >, 111).
  uncondit_formula(111, 69.7041984732828,[[g, 1]])
  uncondit_formula(111, -69.4108863852316,[[si, 1]])
*** On date 910101 formula evaluates to -316.020120674279 ***
jed(1, 3, 11414 <Widget 2 >, 112).
  uncondit_formula(112, 19.8316818243519,[[si, 1]])
*** On date 910101 formula evaluates to 2181.34926021445 ***
jed(1, 4, 31501 <CCA Gain above General Prices >, 113).
  uncondit_formula(113, 19.9154852780808,[[g, 1]])
  uncondit_formula(113, -19.8316818243519,[[si, 1]])
*** On date 910101 formula evaluates to -90.291463049798 ***
jed(1, 5, 21107 <Supplier 1 >, 114).
  uncondit_formula(114, -9000,[])
*** On date 910101 formula evaluates to -9000 ***
jed(1, 6, 31401 <Gearing Gain >, 115).
  uncondit_formula(115, -89.6196837513635,[[g, 1]])
  uncondit_formula(115, 9000,[])

Exhibit 10.1, page 19
*** On date 910101 formula evaluates to -409.760087240953 ***

jeh(2, 900301, 3, Sale of 500 Widget 1’s to Customer 1)

jed(2, 1, 11208 <Customer 1 >, 120).
uncondit_formula(120, 5000,[])
*** On date 910101 formula evaluates to 5000 ***

jed(2, 2, 31401 <Gearing Gain >, 121).
uncondit_formula(121, 49.5994024986421,[[g, 1]])
uncondit_formula(121, -5000,[])
*** On date 910101 formula evaluates to 207.76751765341 ***

jed(2, 3, 32100 <Sales Revenue >, 122).
uncondit_formula(122, -49.5994024986421,[[g, 1]])
*** On date 910101 formula evaluates to -5207.76751765341 ***

jed(2, 4, 33100 <Cost of Sales >, 123).
uncondit_formula(123, 34.9835014691453,[[g, 1]])
*** On date 910101 formula evaluates to 3673.1479275142 ***

jed(2, 5, 11413 <Widget 1 >, 124).
uncondit_formula(124, -34.7054431926158,[[s, 1]])
*** On date 910101 formula evaluates to -3817.36120537529 ***

jed(2, 6, 31501 <CCA Gain above General Prices >, 125).
uncondit_formula(125, -34.9835014691453,[[g, 1]])
uncondit_formula(125, 34.7054431926158,[[s, 1]])
*** On date 910101 formula evaluates to 144.213275623872 ***

jeh(3, 900323, 4, Purchase of 1990 Ford Delivery Truck from Supplier 2)

jed(3, 1, 12801 <1990 Ford Delivery Truck >, 129).
uncondit_formula(129, 357.081157091203,[[s, 1]])
*** On date 910101 formula evaluates to 39276.4831927144 ***

jed(3, 2, 31501 <CCA Gain above General Prices >, 130).
uncondit_formula(130, 360.997156783105,[[g, 1]])
uncondit_formula(130, -357.081157091203,[[s, 1]])
*** On date 910101 formula evaluates to -1373.01717592613 ***

jed(3, 3, 33407 <Depreciation Expense - 1990 Ford Truck >, 131).
formula(131, 131,<, 132, 133, 134)
uncondit_formula(131, -32953,[])
uncondit_formula(131, 1,[[d, 1]])
uncondit_formula(132, 3652.5,[])
uncondit_formula(133, -3221.60037498326,[[s, 1]])
uncondit_formula(133, .097763492701219,[[d, 1],[s, 1]])
uncondit_formula(134, 357.081157091203,[[s, 1]])
*** On date 910101 formula evaluates to 3053.94147207029 ***

jed(3, 4, 12901 <Accum Depr - 1990 Ford Truck >, 132).
formula(132, 135,<, 136, 137, 138)
uncondit_formula(135, -32953,[])
uncondit_formula(135, 1,[[d, 1]])
uncondit_formula(136, 3652.5,[])
uncondit_formula(137, 3221.60037498326,[[s, 1]])
uncondit_formula(137, -.097763492701219,[[d, 1],[s, 1]])
uncondit_formula(138, 357.081157091203,[[s, 1]])

Exhibit 10.1, page 20
**Exhibit 10.1, page 21**

*** On date 910101 formula evaluates to -3053.94147207029 ***

```
jed(3, 5, 21108 <Supplier 2>, 133).
uncondit_formula(139, -36500, []])
```

*** On date 910101 formula evaluates to -36500 ***

```
jed(3, 6, 31401 <Gearing Gain>, 134).
uncondit_formula(140, -360.997156783105, [g, 1])
uncondit_formula(140, 36500, []])
```

*** On date 910101 formula evaluates to -1403.46601678827 ***

```
jed(4, 900401, 5, Pay Supplier 1 $4,000)
uncondit_formula(151, 4000, []])
```

*** On date 910101 formula evaluates to 4000 ***

```
jed(4, 2, 31401 <Gearing Gain>, 140).
uncondit_formula(152, 39.5131845841786, [g, 1])
uncondit_formula(152, -4000, []])
```

*** On date 910101 formula evaluates to 148.74915483431 ***

```
jed(4, 3, 11103 <Bank 1>, 141).
uncondit_formula(153, -4000, []])
```

*** On date 910101 formula evaluates to -4000 ***

```
jed(4, 4, 31401 <Gearing Gain>, 142).
uncondit_formula(154, -39.5131845841786, [g, 1])
uncondit_formula(154, 4000, []])
```

*** On date 910101 formula evaluates to -148.74915483431 ***

```
jed(5, 900501, 6, Declaration of dividend of $0.02 per share)
uncondit_formula(155, 5.90303030303031, [g, 1])
```

*** On date 910101 formula evaluates to 619.797979797971 ***

```
jed(5, 2, 31800 <Retained Earnings>, 144).
uncondit_formula(156, 1.96767676767677, [g, 1])
```

*** On date 910101 formula evaluates to 206.599326599325 ***

```
jed(5, 3, 21502 <Dividend Payable - SH1>, 145).
uncondit_formula(157, -600, [g, 1])
```

*** On date 910101 formula evaluates to -600 ***

```
jed(5, 4, 31401 <Gearing Gain>, 146).
uncondit_formula(158, -5.90303030303031, [g, 1])
```

*** On date 910101 formula evaluates to -19.7979797979715 ***

```
jed(5, 5, 21503 <Dividend Payable - SH2>, 147).
uncondit_formula(159, -200, [g, 1])
```

*** On date 910101 formula evaluates to -200 ***

```
jed(5, 6, 31401 <Gearing Gain>, 148).
uncondit_formula(160, -1.96767676767677, [g, 1])
```

*** On date 910101 formula evaluates to -6.59932659932474 ***

```
jed(6, 900601, 7, Pay office wages of $500 for work in May)
uncondit_formula(161, 11800 <Prepaid Wages>, 152).
```

Exhibit 10.1, page 21
uncondit_formula(164, 3.78607223086227, [[sw, 1]])
*** On date 910101 formula evaluates to 511.093831000513 ***
jed(6, 2, 31501 <CCA Gain above General Prices >, 153).
uncondit_formula(165, 4.89873927038627, [[g, 1]])
uncondit_formula(165, -3.78607223086227, [[sw, 1]])
*** On date 910101 formula evaluates to 3.25702736858148 ***
jed(6, 3, 33401 <Office Wages Expense >, 154).
uncondit_formula(166, 4.89873927038627, [[g, 1]])
*** On date 910101 formula evaluates to 514.350858369093 ***
jed(6, 4, 11800 <Prepaid Wages >, 155).
uncondit_formula(167, -3.78607223086227, [[sw, 1]])
*** On date 910101 formula evaluates to -511.093831000513 ***
jed(6, 5, 31501 <CCA Gain above General Prices >, 156).
uncondit_formula(168, -4.89873927038627, [[g, 1]])
uncondit_formula(168, 3.78607223086227, [[sw, 1]])
*** On date 910101 formula evaluates to -3.25702736858148 ***
jed(6, 6, 11103 <Bank 1 >, 157).
uncondit_formula(169, -500, [])
*** On date 910101 formula evaluates to -500 ***
jed(6, 7, 31401 <Gearing Gain >, 158).
uncondit_formula(170, -4.89873927038627, [[g, 1]])
uncondit_formula(170, 500, [])
*** On date 910101 formula evaluates to -14.3508583690927 ***
jeh(7, 900701, 8, Donation to Red Cross).
jed(7, 1, 11103 <Bank 1 >, 161).
uncondit_formula(173, -100, [])
*** On date 910101 formula evaluates to -100 ***
jed(7, 2, 31401 <Gearing Gain >, 162).
uncondit_formula(174, -975821535329256, [[g, 1]])
uncondit_formula(174, 100, [])
*** On date 910101 formula evaluates to -2.45792145337873 ***
jed(7, 3, 33408 <Donation Expense >, 163).
uncondit_formula(175, 0, [])
uncondit_formula(175, .975821535329256, [[g, 1]])
*** On date 910101 formula evaluates to 102.457921453379 ***

yes

Comment: Closing revenue accounts and a listing of the journal entry generated. (CC/CPP accounting, FIFO inventory flow assumption.)

close_revenue_acs(910101).
Entering close_revenue_acs for period ending 910101
yes
list_je(8,910101).
jeh(8, 910101,closing_acs,Closing revenue and expense for period)
jed(8, 1, 31900 <Profit and Loss >, 165).
**Exhibit 10.1, page 23**

```
uncondit_formula(177, 31.8859251195355, [[g, 1]])
*** On date 910101 formula evaluates to 3347.91301391971 ***

jed(8, 2, 31401 <Gearing Gain >, 166).

uncondit_formula(178, -31.8859251195355, [[g, 1]])
*** On date 910101 formula evaluates to -3347.91301391971 ***

jed(8, 3, 31900 <Profit and Loss >, 167).

uncondit_formula(179, -15.5730359984505, [[g, 1]])
*** On date 910101 formula evaluates to -1635.1154840262 ***

jed(8, 4, 31501 <CCA Gain above General Prices >, 168).

uncondit_formula(180, 15.5730359984505, [[g, 1]])
*** On date 910101 formula evaluates to 1635.1154840262 ***

jed(8, 5, 31900 <Profit and Loss >, 169).

uncondit_formula(181, -49.5994024986419, [[g, 1]])
*** On date 910101 formula evaluates to -5207.7651765338 ***

jed(8, 6, 32100 <Sales Revenue >, 170).

uncondit_formula(182, 49.5994024986419, [[g, 1]])
*** On date 910101 formula evaluates to 5207.7651765338 ***

jed(8, 7, 31900 <Profit and Loss >, 171).

uncondit_formula(183, 34.983501469145, [[g, 1]])
*** On date 910101 formula evaluates to 3673.1479297514 ***

jed(8, 8, 33100 <Cost of Sales >, 172).

uncondit_formula(184, -34.983501469145, [[g, 1]])
*** On date 910101 formula evaluates to -3673.1479297514 ***

jed(8, 9, 31900 <Profit and Loss >, 173).

uncondit_formula(185, 4.89873927038624, [[g, 1]])
*** On date 910101 formula evaluates to 514.350858369089 ***

jed(8, 10, 33401 <Office Wages Expense >, 174).

uncondit_formula(186, -3.81019958307702, [[sw, 1]])
*** On date 910101 formula evaluates to -514.350858369089 ***

jed(8, 11, 31501 <CCA Gain above General Prices >, 175).

uncondit_formula(187, -4.89873927038624, [[g, 1]])
uncondit_formula(187, 3.81019958307702, [[sw, 1]])
*** On date 910101 formula evaluates to 0 ***

jed(8, 12, 31900 <Profit and Loss >, 176).

uncondit_formula(188, 29.0861048937074, [[g, 1]])
*** On date 910101 formula evaluates to 3053.94147207028 ***

jed(8, 13, 33407 <Depreciation Expense - 1990 Ford Truck >, 177).
uncondit_formula(189, -27.7648319271607, [[sp, 1]])
*** On date 910101 formula evaluates to -3053.94147207028 ***

jed(8, 14, 31501 <CCA Gain above General Prices >, 178).
uncondit_formula(190, -29.0861048937074, [[g, 1]])
uncondit_formula(190, 27.7648319271607, [[sp, 1]])
*** On date 910101 formula evaluates to 0 ***

jed(8, 15, 31900 <Profit and Loss >, 179).
uncondit_formula(191, -9.75821533529253, [[g, 1]])
*** On date 910101 formula evaluates to -102.457921453378 ***

jed(8, 16, 33408 <Donation Expense >, 180).
uncondit_formula(192, -9.75821533529253, [[g, 1]])
*** On date 910101 formula evaluates to -102.457921453378 ***
```
Comment: Trial Balance for CC/CPP accounting, FIFO inventory flow assumption, after closing revenue and expense accounts to Profit and Loss.

tb(910101).

Trial Balance for REEFA LTD, date 910101, for Current Cost Accounting, fifo

GL_AC = 11103 <Bank 1 >,
  uncondit_formula(106, 95400,[])
  *** On date 910101 formula evaluates to 95400 ***
  Overall account balance on date 910101 is 95400

GL_AC = 11208 <Customer 1 >,
  uncondit_formula(126, 5000,[])
  *** On date 910101 formula evaluates to 5000 ***
  Overall account balance on date 910101 is 5000

GL_AC = 11413 <Widget 1 >,
  uncondit_formula(116, 34.7054431926158,[[si, 1]])
  *** On date 910101 formula evaluates to 3817.36120537529 ***
  Overall account balance on date 910101 is 3817.36120537529

GL_AC = 11414 <Widget 2 >,
  uncondit_formula(118, 19.8316818243519,[[si, 1]])
  *** On date 910101 formula evaluates to 2181.34926021445 ***
  Overall account balance on date 910101 is 2181.34926021445

GL_AC = 12801 <1990 Ford Delivery Truck >,
  uncondit_formula(141, 357.081157091203,[[sp, 1]])
  *** On date 910101 formula evaluates to 39276.4831927144 ***
  Overall account balance on date 910101 is 39276.4831927144

GL_AC = 12901 <Accum Depr - 1990 Ford Truck >,
  uncondit_formula(137, 147, <, 148, 146, 149)
  uncondit_formula(147, -32953,[])
  uncondit_formula(147, 1,[[d, 1]])
  uncondit_formula(148, 3652.5,[])
  uncondit_formula(146, 3221.60037498326,[[sp, 1]])
  uncondit_formula(146, -0.097763492701219,[[d, 1],[sp, 1]])
  uncondit_formula(149, -357.081157091203,[[sp, 1]])
  *** On date 910101 formula evaluates to -3053.94147207029 ***
  Overall account balance on date 910101 is -3053.94147207029

GL_AC = 21107 <Supplier 1 >,
  uncondit_formula(119, -5000,[])
  *** On date 910101 formula evaluates to -5000 ***
  Overall account balance on date 910101 is -5000

GL_AC = 21108 <Supplier 2 >,
  uncondit_formula(150, -36500,[])
  *** On date 910101 formula evaluates to -36500 ***
  Overall account balance on date 910101 is -36500

GL_AC = 21502 <Dividend Payable - SH1 >,
uncondit_formula(162, -600, [])

*** On date 9/10/01 formula evaluates to -600 ***
Overall account balance on date 9/10/01 is -600

GL_AC = 21503 <Dividend Payable - SH2 >,
uncondit_formula(163, -200, [])

*** On date 9/10/01 formula evaluates to -200 ***
Overall account balance on date 9/10/01 is -200

GL_AC = 31101 <Issued Capital >,
uncondit_formula(108, -806.73660960795, [[g, 1]])

*** On date 9/10/01 formula evaluates to -84704.5831032568 ***
Overall account balance on date 9/10/01 is -84704.5831032568

GL_AC = 31102 <Share Premium Reserve >,
uncondit_formula(109, -193.263390392051, [[g, 1]])

*** On date 9/10/01 formula evaluates to -20291.9945832583 ***
Overall account balance on date 9/10/01 is -20291.9945832583

GL_AC = 31401 <Gearing Gain >,
uncondit_formula(107, -58100, [])
uncondit_formula(107, 553.351368970019, [[g, 1]])

*** On date 9/10/01 formula evaluates to 0 ***
Overall account balance on date 9/10/01 is 0

GL_AC = 31501 <CCA Gain above General Prices >,
uncondit_formula(117, -54.5371250169674, [[sÎ, 1]])
uncondit_formula(117, 3.81019958307702, [[sw, 1]])
uncondit_formula(117, 397.221530899682, [[g, 1]])
uncondit_formula(117, -329.316325164044, [[sp, 1]])

*** On date 9/10/01 formula evaluates to 0 ***
Overall account balance on date 9/10/01 is 0

GL_AC = 31800 <Retained Earnings >,
uncondit_formula(161, 7.87070707070706, [[g, 1]])

*** On date 9/10/01 formula evaluates to 826.397306397295 ***
Overall account balance on date 9/10/01 is 826.397306397295

GL_AC = 31900 <Profit and Loss >,
uncondit_formula(193, 36.6576537892108, [[g, 1]])

*** On date 9/10/01 formula evaluates to 3848.92819388425 ***
Overall account balance on date 9/10/01 is 3848.92819388425

GL_AC = 32100 <Sales Revenue >,
uncondit_formula(127, -2.2737367544323E-0013, [[g, 1]])

*** On date 9/10/01 formula evaluates to -2.387345777544E-0011 ***
Overall account balance on date 9/10/01 is -2.387345777544E-0011

GL_AC = 33100 <Cost of Sales >,
uncondit_formula(128, 2.2737367544323E-0013, [[g, 1]])

*** On date 9/10/01 formula evaluates to 2.387345777544E-0011 ***
Overall account balance on date 9/10/01 is 2.387345777544E-0011

GL_AC = 33401 <Office Wages Expense >,
uncondit_formula(172, 4.89873927038627, [[g, 1]])
uncondit_formula(172, -3.81019958307702, [[sw, 1]])

*** On date 9/10/01 formula evaluates to 3.6379788070917E-0012 ***
Overall account balance on date 9/10/01 is 3.6379788070917E-0012

GL_AC = 33407 <Depreciation Expense -  1990 Ford Truck >,
on date 910101 formula evaluates to 3053.94147207029 **

overall account balance on date 910101 is 1.4551915228367E-011

overall ledger balance on date 910101 is 0

=================================================================

yes
Comment: Money Trial Balance version of the above, i.e., CC/CPP accounting, FIFO inventory flow assumption, after "closing" revenue and expense accounts to Profit and Loss.

money_tb(910101).
MONEY TRIAL BALANCE, REEFA LTD, as at 910101 for Current Cost Accounting, fifo

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance (cents)</th>
<th>CR Balance (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11103</td>
<td>Bank 1</td>
<td>9540000</td>
<td></td>
</tr>
<tr>
<td>11208</td>
<td>Customer 1</td>
<td>500000</td>
<td></td>
</tr>
<tr>
<td>11413</td>
<td>Widget 1</td>
<td>381736</td>
<td></td>
</tr>
<tr>
<td>11414</td>
<td>Widget 2</td>
<td>218135</td>
<td></td>
</tr>
<tr>
<td>12801</td>
<td>1990 Ford Delivery Truck</td>
<td></td>
<td>3927648</td>
</tr>
<tr>
<td>12901</td>
<td>Accum Depr - 1990 Ford Truck</td>
<td></td>
<td>-305394</td>
</tr>
<tr>
<td>21107</td>
<td>Supplier 1</td>
<td></td>
<td>-500000</td>
</tr>
<tr>
<td>21108</td>
<td>Supplier 2</td>
<td></td>
<td>-3650000</td>
</tr>
<tr>
<td>21502</td>
<td>Dividend Payable - SH1</td>
<td></td>
<td>-80000</td>
</tr>
<tr>
<td>21503</td>
<td>Dividend Payable - SH2</td>
<td></td>
<td>-20000</td>
</tr>
<tr>
<td>31101</td>
<td>Issued Capital</td>
<td></td>
<td>-8470458</td>
</tr>
<tr>
<td>31102</td>
<td>Share Premium Reserve</td>
<td></td>
<td>-2029199</td>
</tr>
<tr>
<td>31800</td>
<td>Retained Earnings</td>
<td>82640</td>
<td></td>
</tr>
<tr>
<td>31900</td>
<td>Profit and Loss</td>
<td>384893</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>15035052</td>
<td>-15035051</td>
</tr>
</tbody>
</table>

************************ ERROR ****** ******************** *******

TRIAL BALANCE DOES NOT BALANCE.
Difference is 1 cents.

yes
RUN 3, HC, Standard Cost, Table 9.2 data

Comment: The command below, mv(900601,900701), re-runs the Interpreter
to process all Exchange Events from a new database, the Manufacturing
database, from June 1, 1990 through July 1, 1990, in batch mode. Historical
cost accounting with standard costing is used for this run with the
manufacturing data.

The Exchange Events are defined in manuf_tables, lines 582-635. Since a new
database is being used, Diagnostic Output begins with a causal double-entry
analysis. However, most values come from the standard costs assumed
calculated exogenously from this system, so the value "column" for most of
the ed(..) predicates below is "_1", indicating "undefined".

mv(900601,900701).
Beginning initialization
closing_dates(900601, 900701).

Now using money values exchanged to value non-monetary assets and obligations
except for the following values accepted from input:
Eno 0 Pay Factory Rent Lno 1 V= 1000
Eno 1 Cash Purchase Lno 1 V= 20100
Eno 1 Cash Purchase Lno 2 V= 440
Eno 11 Record Factory Wages Lno 1 V= 445
Eno 11 Record Factory Wages Lno 2 V= 160
Generation of exchange money values complete.

ed( 0, 1, 70, 1, 1000).
ed( 0, 2, 1,-1000,-1000).
ed( 1, 1, 31, 500, 20100).
ed( 1, 2, 32, 2000, 440).
ed( 1, 3, 1,-20540,-20540).
ed( 2, 1, 420, 40,_1).
ed( 2, 2, 31,-40,_1).
ed( 3, 1, 421, 79,_1).
ed( 3, 2, 510,-79,_1).
ed( 4, 1, 430, 10000,_1).
ed( 4, 2, 429,-10000,_1).
ed( 5, 1, 431, 825,_1).
ed( 5, 2, 32,-825,_1).
ed( 6, 1, 432, 24,_1).
ed( 6, 2, 511,-24,_1).
ed( 7, 1, 51, 820,_1).
ed( 7, 2, 439,-820,_1).
ed( 8, 1, 420,-1,_1).
ed( 8, 2, 31, 1,_1).
ed( 9, 1, 429, 10000,_1).
ed( 9, 2, 420,-39,_1).

Exhibit 10.1, page 28
Here are the present values of the Multiview Interpreter's switches
---------------------------------------------------------------------
\text{price\_level\_accounting\_switch}(HistoricalCost).
\text{fifo\_lifo\_switch}(standard\_cost).

Change switches? Enter y. or n. (in lower case followed by a dot): n.

\begin{quote}
\textit{Comment}: For standard costing, the prototype Interpreter calculates standard
\textit{costs for product resources from standard costs of inputs\textsuperscript{7}. Five
\textit{derived\_unit\_standard\_costs are listed below. (If a predicate won't fit onto
one line, CYBER Prolog lists it over multiple lines -- hence five predicates
take 30 lines to list.)}

Dates on which standards have been set are: [900101]
Rnos of resources for which standard costs are to be derived are: [1030, 1031,
1050, 1051, 1053]
\text{derived\_unit\_standard\_cost}(1030,
\phantom{\text{derived\_unit\_standard\_cost}}900101,
\phantom{\text{derived\_unit\_standard\_cost}}.289999999999999,
\phantom{\text{derived\_unit\_standard\_cost}}.16,
\phantom{\text{derived\_unit\_standard\_cost}}.05,
\phantom{\text{derived\_unit\_standard\_cost}}.0800000000000001).
\end{quote}

\text{\textit{i.e.}, derived\_unit\_standard\_cost}(1030, 900101, .29, .16, .05, .08)

\text{derived\_unit\_standard\_cost}(1031,
\phantom{\text{derived\_unit\_standard\_cost}}900101,
\phantom{\text{derived\_unit\_standard\_cost}}.539999999999999,
\phantom{\text{derived\_unit\_standard\_cost}}.279999999999999,
\phantom{\text{derived\_unit\_standard\_cost}}.539999999999999,
\phantom{\text{derived\_unit\_standard\_cost}}.279999999999999,
\phantom{\text{derived\_unit\_standard\_cost}}.279999999999999,
\textsuperscript{7} This is invoked in Interpreter, line 125, which calls the procedure in lines 1205-
1294. Simple material plus labour plus overhead rules based on data from the
\textit{composite\_resource table} (obtained in line 1249) are followed.
Comment: Below, the Interpreter works through the eleven Exchange Events from Table 9.2. Settings are Historical Cost, and Standard Costing. (The Interpreter was only coded for Standard Costing for manufacturing events.)

Elapsed time to generate the journal entries on the next 6 pages was 22 sec.

Here are the values of the Multiview Interpreter's switches for this run

for accounting period from 900601 to Report date 900701:

Run date and time started: "1991-01-29" "22:56:45.656"

price_level_accounting_switch(HistoricalCost).

fifo_lifo_switch(standard_cost).

---- Period starting 900601 and finishing 900701 ----

CALLING simple for event 0 of type Pay Factory Rent
simple called for event 0

Starting straight line depreciation for Rno= 1501

CALLING simple for event 1 of type Cash Purchase
simple called for event 1

CALLING manuf_event for event 2 of type Raw Materials Issue
manuf_event called for event 2

CALLING manuf_event for event 3 of type Record Direct Labour
manuf_event called for event 3

CALLING manuf_event for event 4 of type Interdepartmental Goods Transfer
manuf_event called for event 4

Exhibit 10.1, page 30
CALLING manuf_event for event 5 of type Raw Materials Issue
manuf_event called for event 5
CALLING manuf_event for event 6 of type Record Direct Labour
manuf_event called for event 6
CALLING manuf_event for event 7 of type Transfer to Finished Goods
manuf_event called for event 7
CALLING manuf_event for event 8 of type Return to RM Store
manuf_event called for event 8
CALLING production_summary for event 9 of type Production Summary
production_summary called for event 9
CALLING manuf_event for event 10 of type Recognize Idle Time
manuf_event called for event 10
CALLING manuf_event for event 11 of type Record Factory Wages
manuf_event called for event 11
ALL EVENTS UP TO REPORT DATE NOW PROCESSED
Run date and time completed: "1991-01-29" "22:57:07.203"

**Comment:** Trial Balance for HC accounting, standard costing, Table 9.2 data:

Trial Balance for REEFA LTD, date 900701, for Historical Cost, standard_cost

<table>
<thead>
<tr>
<th>GL_AC</th>
<th>Description</th>
<th>Formula Details</th>
</tr>
</thead>
</table>
| 2000 <Cash at Bank A> | uncondit_formula(127,-21540,[]) | *** On date 900701 formula evaluates to -21540 ***
|             | Overall account balance on date 900701 is -21540 |
| 2021 <Polyethylene Pellets> | uncondit_formula(134, 18440,[]) | *** On date 900701 formula evaluates to 18440 ***
|             | Overall account balance on date 900701 is 18440 |
| 2022 <Empty Cartons> | uncondit_formula(135, 258.5,[]) | *** On date 900701 formula evaluates to 258.5 ***
|             | Overall account balance on date 900701 is 258.5 |
| 2071 <Carton 1 Litre Bottles> | uncondit_formula(158, 3132.39999999998,[]) | *** On date 900701 formula evaluates to 3132.39999999998 ***
|             | Overall account balance on date 900701 is 3132.39999999998 |
| 2090 <Prepaid Rent> | uncondit_formula(118, 1000,[]) | *** On date 900701 formula evaluates to 1000 ***
|             | formula(114, 124, 125, 123, 126) |
|             | uncondit_formula(124, -33023,[]) |
|             | uncondit_formula(124, 1, [d, 1]) |
|             | uncondit_formula(125, 30.4375,[]) |
|             | uncondit_formula(123, 1084944.55852155,[]) |
|             | uncondit_formula(123, -32.8542094455852, [d, 1]) |
|             | uncondit_formula(126, -1000,[]) |
|             | *** On date 900701 formula evaluates to -985.626283369958 *** |

Exhibit 10.1, page 31
Overall account balance on date 900701 is 14.3737166300416

GL_AC = 2311 <Factory Wages Payable >,
uncondit_formula(183,-605,[])
*** On date 900701 formula evaluates to -605 ***
Overall account balance on date 900701 is -605

GL_AC = 2710 <Profit and Loss >,
uncondit_formula(170,-67.0999999999976,[])
*** On date 900701 formula evaluates to -67.0999999999976 ***
Overall account balance on date 900701 is -67.0999999999976

GL_AC = 2802 <Production OH - General >,
formula(113,120,<,121,119,122)
uncondit_formula(120,-33023,[])
uncondit_formula(119,-108494.55852155,[])
uncondit_formula(119,32.8542094455852,[]) uncondit_formula(121,30.4375,[])
uncondit_formula(119,-108494.55852155,[])
uncondit_formula(122,1000,[])
*** On date 900701 formula evaluates to 985.626283369958 ***
Overall account balance on date 900701 is 985.626283369958

GL_AC = 2803 <Production OH - Contra >,
uncondit_formula(107,-1,[[gl_account(2802)],[1]])
*** On date 900701 formula evaluates to -985.626283369958 ***
Overall account balance on date 900701 is -985.626283369958

GL_AC = 2810 <Overhead Incurred - Moulding >,
uncondit_formula(104,.6,[[gl_account(2802),1]])
uncondit_formula(104,20,[])
*** On date 900701 formula evaluates to 611.375770021976 ***
Overall account balance on date 900701 is 611.375770021976

GL_AC = 2811 <Overhead Applied - Moulding >,
uncondit_formula(105,-799.999999999993,[])
*** On date 900701 formula evaluates to -799.999999999993 ***
Overall account balance on date 900701 is -799.999999999993

GL_AC = 2820 <Overhead Incurred - Packing >,
uncondit_formula(105,.3,[[gl_account(2802),1]])
uncondit_formula(105,44,[])
*** On date 900701 formula evaluates to 339.687885010988 ***
Overall account balance on date 900701 is 339.687885010988

GL_AC = 2821 <Overhead Applied - Packing >,
uncondit_formula(159,-32.7999999999997,[])
*** On date 900701 formula evaluates to -32.7999999999997 ***
Overall account balance on date 900701 is -32.7999999999997

GL_AC = 2830 <Office Share of Rent >,
uncondit_formula(106,.1,[[gl_account(2802),1]])
*** On date 900701 formula evaluates to 98.5626283369957 ***
Overall account balance on date 900701 is 98.5626283369957

---

8 This formula, which is based on the balance in account 2802, was not generated as a result of any Exchange Event. See footnote to journal entry 0 in two pages.

Exhibit 10.1, page 32
Comment: Money Trial Balance for HC accounting, standard costing, Table 9.2 data:

MONEY TRIAL BALANCE, REEFA LTD, as at 900701
for Historical Cost, standard_cost

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance (cents)</th>
<th>CR Balance (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>Cash at Bank A</td>
<td></td>
<td>-2154000</td>
</tr>
<tr>
<td>2021</td>
<td>Polyethylene Pellets</td>
<td>1844000</td>
<td></td>
</tr>
<tr>
<td>2022</td>
<td>Empty Cartons</td>
<td>25850</td>
<td></td>
</tr>
<tr>
<td>2071</td>
<td>Carton 1 Litre Bottles</td>
<td>313240</td>
<td></td>
</tr>
<tr>
<td>2090</td>
<td>Prepaid Rent</td>
<td>1437</td>
<td></td>
</tr>
<tr>
<td>2311</td>
<td>Factory Wages Payable</td>
<td></td>
<td>-60500</td>
</tr>
<tr>
<td>2710</td>
<td>Profit and Loss</td>
<td></td>
<td>-6710</td>
</tr>
<tr>
<td>2802</td>
<td>Production OH - General</td>
<td>98563</td>
<td></td>
</tr>
<tr>
<td>2803</td>
<td>Production OH - Contra</td>
<td></td>
<td>-98563</td>
</tr>
<tr>
<td>2810</td>
<td>Overhead Incurred - Moulding</td>
<td>61138</td>
<td></td>
</tr>
<tr>
<td>2811</td>
<td>Overhead Applied - Moulding</td>
<td></td>
<td>-80000</td>
</tr>
<tr>
<td>2820</td>
<td>Overhead Incurred - Packing</td>
<td>33969</td>
<td></td>
</tr>
<tr>
<td>2821</td>
<td>Overhead Applied - Packing</td>
<td></td>
<td>-3280</td>
</tr>
<tr>
<td>2830</td>
<td>Office Share of Rent</td>
<td>9856</td>
<td></td>
</tr>
<tr>
<td>3001</td>
<td>Materials Price Variance</td>
<td>10000</td>
<td></td>
</tr>
<tr>
<td>3002</td>
<td>Labour Rate Variance</td>
<td>5000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2403053</td>
<td>-2403053</td>
</tr>
</tbody>
</table>

Exhibit 10.1, page 33
Journal entries for REEFA LTD, for period from 900601 to Report date 900701 for Historical Cost, standard_cost

jeh(0, 900601, -1, Test cost allocation)

jed(0, 1, 2810 <Overhead Incurred - Moulding >, 100).
    uncondit_formula(100, .600000000000001, [[gl_account(2802), 1]])
    *** On date 900701 formula evaluates to 591.375770021976 ***

jed(0, 2, 2820 <Overhead Incurred - Packing >, 101).
    uncondit_formula(101, .300000000000001, [[gl_account(2802), 1]])
    *** On date 900701 formula evaluates to 295.687885010988 ***

jed(0, 3, 2830 <Office Share of Rent >, 102).
    uncondit_formula(102, .1, [[gl_account(2802), 1]])
    *** On date 900701 formula evaluates to 98.5626283369957 ***

jed(0, 4, 2803 <Production OH - Contra >, 103).
    uncondit_formula(103, -1, [[gl_account(2802), 1]])
    *** On date 900701 formula evaluates to -985.626283369958 ***

jeh(1, 900601, 0, Pay Factory Rent)

jed(1, 1, 2090 <Prepaid Rent >, 108).
    uncondit_formula(108, 1000, [])
    *** On date 900701 formula evaluates to 1000 ***

jed(1, 2, 2802 <Production OH - General formula(109, 109, <, 110, 111, 112)
    uncondit_formula(109, -33023, [])
    uncondit_formula(109, 1, [[d, 1]])
    uncondit_formula(110, 30.4375, [])
    uncondit_formula(111, -1084944.55852155, [])
    uncondit_formula(111, 32.8542094455852, [[d, 1]])
    uncondit_formula(112, 1000, [])
    *** On date 900701 formula evaluates to 985.626283369958 ***

jed(1, 3, 2090 <Prepaid Rent >, 110).
    formula(110, 113, <, 114, 115, 116)
    uncondit_formula(113, -33023, [])
    uncondit_formula(113, 1, [[d, 1]])
    uncondit_formula(114, 30.4375, [])
    uncondit_formula(115, 1084944.55852155, [])
    uncondit_formula(115, -32.8542094455852, [[d, 1]])
    uncondit_formula(116, -1000, [])
    *** On date 900701 formula evaluates to -985.626283369958 ***

---

Comment: Journal Entries for HC accounting, standard costing, Table 9.2 data:

This journal entry is not a response to any Exchange Event in Table 9.2. It would be entered through a terminal as a direct-entry journal entry. For this test it was hard-coded into the Interpreter on lines 132-140, and posted on line 141. The valuation routine (in Formula, line 160-162) uses the current balance in account 2802.
jed(1, 4, 2000 <Cash at Bank A >, 111).

uncondit_formula(117,-1000,[])
*** On date 900701 formula evaluates to -1000 ***

jeh(2, 900601, 1,Purchase of Raw Materials)

ejd(2, 1, 3001 <Materials Price Variance >, 116).
uncondit_formula(128, 100,[])
*** On date 900701 formula evaluates to 100 ***

jed(2, 2, 2021 <Polyethelene Pellets >, 117).
uncondit_formula(129, 20000,[])
*** On date 900701 formula evaluates to 20000 ***

jed(2, 3, 3001 <Materials Price Variance >, 118).
uncondit_formula(130, 1.8189894035459E-0012,[])
*** On date 900701 formula evaluates to 1.8189894035459E-0012 ***

jed(2, 4, 2022 <Empty Cartons >, 119).
uncondit_formula(131, 439.999999999998,[])
*** On date 900701 formula evaluates to 439.999999999998 ***

jeh(3, 900606, 2,Issue 40 x 20 kg bags of polyethelene pellets to Job 123)

jed(3, 1, 2040 <Materials in Progress, Job 123, Step 1 >, 124).
uncondit_formula(136, 1600,[])
*** On date 900701 formula evaluates to 1600 ***

jed(3, 2, 2021 <Polyethelene Pellets >, 125).
uncondit_formula(137,-1600,[])
*** On date 900701 formula evaluates to -1600 ***

jeh(4, 900606, 3,79 hrs factory moulding labour on Job 123)

jed(4, 1, 2041 <Labour in Progress, Job 123, Step 1 >, 127).
uncondit_formula(139, 395,[])
*** On date 900701 formula evaluates to 395 ***

jed(4, 2, 3000 <Labour Clearing >, 128).
uncondit_formula(140,-395,[])
*** On date 900701 formula evaluates to -395 ***

jeh(5, 900606, 4,Transfer of 10,000 1L bottles to packing)

jed(5, 1, 2045 <Materials in Progress, Job 123, Step 2 >, 131).
uncondit_formula(143, 2899.99999999999,[])
*** On date 900701 formula evaluates to 2899.99999999999 ***

jed(5, 2, 2040 <Materials in Progress, Job 123, Step 1 >, 132).
uncondit_formula(144, -1599.99999999999,[])
*** On date 900701 formula evaluates to -1599.99999999999 ***

jed(5, 3, 2041 <Labour in Progress, Job 123, Step 1 >, 133).
uncondit_formula(145, -499.999999999998,[])
*** On date 900701 formula evaluates to -499.999999999998 ***

uncondit_formula(146, -799.999999999993,[])

Exhibit 10.1, page 35
*** On date 900701 formula evaluates to -799.999999999993 ***

jeh(6, 900607, 5, Issue 825 cartons to Job 123)
jed(6, 1, 2045 <Materials in Progress, Job 123, Step 2 >, 137).
uncondit_formula(149, 181.499999999999, [])
*** On date 900701 formula evaluates to 181.499999999999 ***
jed(6, 2, 2022 <Empty Cartons >, 138).
uncondit_formula(150, -181.499999999999, [])
*** On date 900701 formula evaluates to -181.499999999999 ***

jeh(7, 900607, 6, 24 hrs factory packing labour on Job 123)
jed(7, 1, 2046 <Labour in Progress, Job 123, Step 2 >, 139).
uncondit_formula(151, 96, [])
*** On date 900701 formula evaluates to 96 ***
jed(7, 2, 3000 <Labour Clearing >, 140).
uncondit_formula(152, -96, [])
*** On date 900701 formula evaluates to -96 ***

jeh(8, 900607, 7, Transfer 820 cartons of 1L bottles to Finished Goods)
jed(8, 1, 2071 <Carton 1 Litre Bottles >, 142).
uncondit_formula(154, 3132.39999999998, [])
*** On date 900701 formula evaluates to 3132.39999999998 ***
jed(8, 2, 2045 <Materials in Progress, Job 123, Step 2 >, 143).
uncondit_formula(155, -3033.99999999999, [])
*** On date 900701 formula evaluates to -3033.99999999999 ***
jed(8, 3, 2046 <Labour in Progress, Job 123, Step 2 >, 144).
uncondit_formula(156, -65.5999999999995, [])
*** On date 900701 formula evaluates to -65.5999999999995 ***
jed(8, 4, 2821 <Overhead Applied - Packing >, 145).
uncondit_formula(157, -32.7999999999997, [])
*** On date 900701 formula evaluates to -32.7999999999997 ***

jeh(9, 900607, 8, Return of 1 20kg bag of pellets to store)
jed(9, 1, 2040 <Materials in Progress, Job 123, Step 1 >, 148).
uncondit_formula(160, -40, [])
*** On date 900701 formula evaluates to -40 ***
jed(9, 2, 2021 <Polyethene Pellets >, 149).
uncondit_formula(161, 40, [])
*** On date 900701 formula evaluates to 40 ***

jeh(10, 900607, 9, Completion of production of Job 123 - Moulding and Packing)
jed(10, 1, 2040 <Materials in Progress, Job 123, Step 1 >, 150).
uncondit_formula(162, 39.9999999999854, [])
*** On date 900701 formula evaluates to 39.9999999999854 ***
jed(10, 2, 2710 <Profit and Loss >, 151).
uncondit_formula(163, -39.9999999999854, [])
*** On date 900701 formula evaluates to -39.9999999999854 ***
jed(10, 3, 2041 <Labour in Progress, Job 123, Step 1 >, 152).
uncondit_formula(164, 104.999999999998, [])

Exhibit 10.1, page 36
**On date 900701 formula evaluates to 104.999999999998***

jed(10, 4, 2710 <Profit and Loss >, 153).
uncondit_formula(165, -104.999999999998,[])

**On date 900701 formula evaluates to -104.999999999998***

jed(10, 5, 2045 <Materials in Progress, Job 123, Step 2 >, 154).
uncondit_formula(166, -47.4999999999854,[])

**On date 900701 formula evaluates to -47.4999999999854***

jed(10, 6, 2710 <Profit and Loss >, 155).
uncondit_formula(167, 47.4999999999854,[])

*** On date 900701 formula evaluates to 47.4999999999854 ***

jed(10, 7, 2046 <Labour in Progress, Job 123, Step 2 >, 156).
uncondit_formula(168, -30.4000000000005,[])

**On date 900701 formula evaluates to -30.4000000000005***

jed(10, 8, 2710 <Profit and Loss >, 157).
uncondit_formula(169, 30.4000000000005,[])

*** On date 900701 formula evaluates to 30.4000000000005 ***

jeh(11, 900606, 10, Idle time Job 123)

jed(11, 1, 2810 <Overhead Incurred - Moulding >, 159).
uncondit_formula(171, 20,[])

*** On date 900701 formula evaluates to 20 ***

jed(11, 2, 3000 <Labour Clearing >, 160).
uncondit_formula(172, -20,[])

*** On date 900701 formula evaluates to -20 ***

jed(11, 3, 2820 <Overhead Incurred - Packing >, 161).
uncondit_formula(173, 44,[])

*** On date 900701 formula evaluates to 44 ***

jed(11, 4, 3000 <Labour Clearing >, 162).
uncondit_formula(174, -44,[])

*** On date 900701 formula evaluates to -44 ***

jeh(12, 900606, 11, Factory wages, employees M,N,Q)

jed(12, 1, 3002 <Labour Rate Variance >, 163).
uncondit_formula(175, 30,[])

*** On date 900701 formula evaluates to 30 ***

jed(12, 2, 3000 <Labour Clearing >, 164).
uncondit_formula(176, 415,[])

*** On date 900701 formula evaluates to 415 ***

jed(12, 3, 3002 <Labour Rate Variance >, 165).
uncondit_formula(177, 20,[])

*** On date 900701 formula evaluates to 20 ***

jed(12, 4, 3000 <Labour Clearing >, 166).
uncondit_formula(178, 140,[])

*** On date 900701 formula evaluates to 140 ***

jed(12, 5, 2311 <Factory Wages Payable >, 167).
uncondit_formula(179, -220,[])

*** On date 900701 formula evaluates to -220 ***

jed(12, 6, 2311 <Factory Wages Payable >, 168).
uncondit_formula(180, -225,[])

Exhibit 10.1, page 37
*** On date 900701 formula evaluates to -225 ***

\[ \text{jed(12, 7, 2311 < Factory Wages Payable, 169).} \]

uncondit_formula(181,-160,[])

*** On date 900701 formula evaluates to -160 ***

\[ \text{yes} \]
RUN 4, CC/CPP, Standard Costing, Table 9.2 data

Comment: The command below, mv(900601,900701), re-runs the Interpreter to process again all Exchange Events from Table 9.2 from June 1, 1990 through July 1, 1990, this time for CC/CPP accounting, not historical cost.

Diagnostic Output commences on this page. The Interpreter has not repeated causal double-entry reasoning, since the facts are unchanged from Run 3. Similarly, it was not necessary to recompute derived standard costs.
CC/CPP, Standard Costing accounting rules are selected in the lower half of this page. Linear interpolation was used for index values for the dates in June, 1990.

mv(900601,900701).
Beginning initialization

closing_dates(900601, 900701).

Here are the present values of the Multiview Interpreter's switches
-------------------------------------------------------------------
price_level_accounting_switch(HistoricalCost).

fifo_lifo_switch(standard_cost).

Change switches? Enter y, or n. (in lower case followed by a dot): y.
This Interpreter handles Historical Cost (hc),
CPP (cpp) or CC/CPP (cc/cpp) accounting.
Enter choice, either hc., cpp. or cc/cpp. (in lower case followed by a dot): cc/cpp.
This Interpreter handles either fifo (f),
lifo (l) or std. cost (s) inventory valuation.
Enter choice, either f., l. or s. (in lower case followed by a dot): s.

Here are the values of the Multiview Interpreter's switches for this run
--------------------------------------------------------------------------
for accounting period from 900601 to Report date 900701:
--------------------------------------------------------------------------
Run date and time started: "1991-01-29" "22:57:39.768"
price_level_accounting_switch(CurrentCost Accounting).

fifo_lifo_switch(standard_cost).

----- Period starting 900601 and finishing 900701 ------
CALLING simple for event 0 of type Pay Factory Rent
simple called for event 0
Starting straight line depreciation for Rno= 1501

Exhibit 10.1, page 39
CALLING simple for event 1 of type Cash Purchase
simple called for event 1
CALLING manuf_event for event 2 of type Raw Materials Issue
manuf_event called for event 2
CALLING manuf_event for event 3 of type Record Direct Labour
manuf_event called for event 3
CALLING manuf_event for event 4 of type Interdepartmental Goods Transfer
manuf_event called for event 4
CALLING manuf_event for event 5 of type Raw Materials Issue
manuf_event called for event 5
CALLING manuf_event for event 6 of type Record Direct Labour
manuf_event called for event 6
CALLING manuf_event for event 7 of type Transfer to Finished Goods
manuf_event called for event 7
CALLING manuf_event for event 8 of type Return to RM Store
manuf_event called for event 8
CALLING production_summary for event 9 of type Production Summary
production_summary called for event 9
CALLING manuf_event for event 10 of type Recognize Idle Time
manuf_event called for event 10
CALLING manuf_event for event 11 of type Record Factory Wages
manuf_event called for event 11
ALL EVENTS UP TO REPORT DATE NOW PROCESSED
Run date and time completed: "1991-01-29" "23:01:15.540"

Comment:

As shown by the start and end times for this run, the FA journal entries shown on the next 10 pages took over 3 minutes, elapsed time, to generate and post. There are some obvious ways to reduce this run time. Consider, for example, journal entry 4 on page 46. First, the coefficients of lines 1 and 2 exactly equal those in lines 3 and 4. Yet the prototype Interpreter recomputes all coefficients from scratch. Second, the debit and credit in lines 2 and 4 exactly cancel, they achieve nothing. By processing the Exchange Event as a unit, not two independent lines, computation and posting of lines 2 and 4 could have been avoided entirely. Refinements such as these could easily be built into a commercial product.
Trial Balance for REEFA LTD, date 900701, for Current Cost Accounting, standard_cost

GL_AC = 2000 <Cash at Bank A>
    uncondit_formula(128, -21540,[])
*** On date 900701 formula evaluates to -21540 ***
Overall account balance on date 900701 is -21540

GL_AC = 2021 <Polyethylene Pellets>
    uncondit_formula(142, 184.4,[[si, 1]])
*** On date 900701 formula evaluates to 20807.9028037381 ***
Overall account balance on date 900701 is 20807.9028037381

GL_AC = 2022 <Empty Cartons>
    uncondit_formula(143, 2.58499999999998,[[si, 1]])
*** On date 900701 formula evaluates to 291.694299065417 ***
Overall account balance on date 900701 is 291.694299065417

GL_AC = 2071 <Carton 1 Litre Bottles>
    uncondit_formula(182, 31.3239999999994,[[si, 1]])
*** On date 900701 formula evaluates to 3534.63528971953 ***
Overall account balance on date 900701 is 3534.63528971953

GL_AC = 2090 <Prepaid Rent>
    uncondit_formula(119, 9.79747854077254,[[g, 1]])
*** On date 900701 formula evaluates to 1004.02360515021 ***
uncondit_formula(125, -33023,[])
uncondit_formula(126, 30.4375,[])
uncondit_formula(124, 10629.7210300429,[[g, 1]])
uncondit_formula(124, -321884121017167,[[d, 1],[g, 1]])
uncondit_formula(127, -9.79747854077254,[[g, 1]])
*** On date 900701 formula evaluates to -88.6956630843924 ***
Overall account balance on date 900701 is -88.6956630843924

GL_AC = 2311 <Factory Wages Payable>
    uncondit_formula(220, -605,[])
*** On date 900701 formula evaluates to -605 ***
Overall account balance on date 900701 is -605

GL_AC = 2603 <Gearing Gain>
    uncondit_formula(129, -216.961189979739,[[g, 1]])
uncondit_formula(129, 22145,[])
*** On date 900701 formula evaluates to -88.6956630843924 ***
Overall account balance on date 900701 is -88.6956630843924

GL_AC = 2604 <CCA Gain above General Prices>
    uncondit_formula(141, -184.780561627716,[[si, 1]])
uncondit_formula(141, 207.16371143897,[[g, 1]])
uncondit_formula(141, -5.4857512953368,[[sw, 1]])
**Exhibit 10.1, page 42**

---

**On date 900701 formula evaluates to -235.216193863307**

Overall account balance on date 900701 is -235.216193863307

**GL_AC = 2710 <Profit and Loss >,**
- `uncondit_formula(196, 4.90999999999997,[[sw, 1]])`
- `uncondit_formula(196, -5.58099999999936,[[si, 1]])`

**On date 900701 formula evaluates to -80.1700353290871**

Overall account balance on date 900701 is -80.1700353290871

**GL_AC = 2802 <Production OH - General >,**
- `formula(114, 121, <, 122, 120, 123)`
- `uncondit_formula(121, -33023,[[]])`
- `uncondit_formula(121, 1,[[d, 1]])`
- `uncondit_formula(122, 30.4375,[[]])`
- `uncondit_formula(120, -10629.7210300429,[[g, 1]])`
- `uncondit_formula(120, .32188412017167,[[d, 1],[g, 1]])`
- `uncondit_formula(123, 9.79747854077254,[[g, 1]])`

**On date 900701 formula evaluates to 989.592054359615**

Overall account balance on date 900701 is 989.592054359615

**GL_AC = 2803 <Production OH - Contra >,**
- `uncondit_formula(107, -1,[[gl_account(2802), 1]])`

**On date 900701 formula evaluates to -989.592054359615**

Overall account balance on date 900701 is -989.592054359615

**GL_AC = 2810 <Overhead Incurred - Moulding >,**
- `uncondit_formula(104, .600000000000001,[[gl_account(2802), 1]])`
- `uncondit_formula(104, .199999999999999,[[sw, 1]])`

**On date 900701 formula evaluates to 616.142045802579**

Overall account balance on date 900701 is 616.142045802579

**GL_AC = 2811 <Overhead Applied - Moulding >,**
- `uncondit_formula(164, -7.99999999999989,[[si, 1]])`

**On date 900701 formula evaluates to -902.728971962599**

Overall account balance on date 900701 is -902.728971962599

**GL_AC = 2820 <Overhead Incurred - Packing >,**
- `uncondit_formula(105, .300000000000001,[[gl_account(2802), 1]])`
- `uncondit_formula(105, .439999999999996,[[sw, 1]])`

**On date 900701 formula evaluates to 346.128605318871**

Overall account balance on date 900701 is 346.128605318871

**GL_AC = 2821 <Overhead Applied - Packing >,**
- `uncondit_formula(183, -.327999999999992,[[si, 1]])`

**On date 900701 formula evaluates to -37.0118878504661**

Overall account balance on date 900701 is -37.0118878504661

**GL_AC = 2830 <Office Share of Rent >,**
- `uncondit_formula(106, .1,[[gl_account(2802), 1]])`

**On date 900701 formula evaluates to 98.9592054359614**

Overall account balance on date 900701 is 98.9592054359614

**GL_AC = 3000 <Labour Clearing >,**
- `uncondit_formula(154, 3.5527136788005E-0014,[[sw, 1]])`

**On date 900701 formula evaluates to 3.9766968716771E-0012**

Overall account balance on date 900701 is 3.9766968716771E-0012

**GL_AC = 3001 <Materials Price Variance >,**
- `uncondit_formula(140, -19.619438372864,[[si, 1]])`

---

---
On date 900701 formula evaluates to \(-2213.87942903723\)
Overall account balance on date 900701 is \(-2213.87942903723\)

Overall ledger balance on date 900701 is 0

Comment: Money Trial Balance for CC/CPP accounting, standard costing, Table 9.2 data:

MONEY TRIAL BALANCE, REEFA LTD, as at 900701
for Current Cost Accounting, standard_cost

<table>
<thead>
<tr>
<th>Account</th>
<th>Description</th>
<th>DR Balance (cents)</th>
<th>CR Balance (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>Cash at Bank A</td>
<td></td>
<td>-2154000</td>
</tr>
<tr>
<td>2021</td>
<td>Polyethelene Pellets</td>
<td>2080790</td>
<td></td>
</tr>
<tr>
<td>2022</td>
<td>Empty Cartons</td>
<td>29169</td>
<td></td>
</tr>
<tr>
<td>2071</td>
<td>Carton 1 Litre Bottles</td>
<td>353464</td>
<td></td>
</tr>
<tr>
<td>2090</td>
<td>Prepaid Rent</td>
<td>1443</td>
<td></td>
</tr>
<tr>
<td>2311</td>
<td>Factory Wages Payable</td>
<td></td>
<td>-60500</td>
</tr>
<tr>
<td>2603</td>
<td>Gearing Gain</td>
<td></td>
<td>-8870</td>
</tr>
<tr>
<td>2604</td>
<td>CCA Gain above General Prices</td>
<td></td>
<td>-23522</td>
</tr>
<tr>
<td>2710</td>
<td>Profit and Loss</td>
<td></td>
<td>-8017</td>
</tr>
<tr>
<td>2802</td>
<td>Production OH - General</td>
<td>98959</td>
<td>-98959</td>
</tr>
<tr>
<td>2803</td>
<td>Production OH - Contra</td>
<td></td>
<td>-90273</td>
</tr>
<tr>
<td>2810</td>
<td>Overhead Incurred - Moulding</td>
<td>61614</td>
<td>-3701</td>
</tr>
<tr>
<td>2811</td>
<td>Overhead Applied - Moulding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2820</td>
<td>Overhead Incurred - Packing</td>
<td>34613</td>
<td></td>
</tr>
<tr>
<td>2821</td>
<td>Overhead Applied - Packing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2830</td>
<td>Office Share of Rent</td>
<td>9896</td>
<td></td>
</tr>
<tr>
<td>3001</td>
<td>Materials Price Variance</td>
<td></td>
<td>-221388</td>
</tr>
<tr>
<td>3002</td>
<td>Labour Rate Variance</td>
<td></td>
<td>-719</td>
</tr>
</tbody>
</table>

************** ERROR **************

TRIAL BALANCE DOES NOT BALANCE.
Difference is -1 cents.

************** ERROR **************

Exhibit 10.1, page 43
**Comment:** Journal Entries for CC/CPP accounting, standard costing,
Table 9.2 data:

Journal entries for REEFA LTD, for period from 900601 to Report date 900701 for Current Cost Accounting, standard_cost

jeh(0, 900601,.1,Test cost allocation)

jed(0, 1, 2810 <Overhead Incurred - Moulding >, 100).
uncondit_formula(100, .600000000000001,[[gl_account(2802), 1]])
*** On date 900701 formula evaluates to 593.755232615767 ***

jed(0, 2, 2820 <Overhead Incurred - Packing >, 101).
uncondit_formula(101, .300000000000001,[[gl_account(2802), 1]])
*** On date 900701 formula evaluates to 296.877616307884 ***

jed(0, 3, 2830 <Office Share of Rent >, 102).
uncondit_formula(102, .1,[[gl_account(2802), 1]])
*** On date 900701 formula evaluates to 98.9592054359614 ***

jed(0, 4, 2803 <Production OH - Contra >, 103).
uncondit_formula(103,-1,[[gl_account(2802), 1]])
S** On date 900701 formula evaluates to -989.592054359615 * * *

jeh(1, 900601,0,Pay Factory Rent)

jed(1, 1, 2090 <Prepaid Rent >, 108).
uncondit_formula(108, 9.79747854077254,[[g, 1]])
*** On date 900701 formula evaluates to 1004.02360515021 ***

jed(1, 2, 2802 <Production OH - General >, 109).
formula(109, 109,<, 110, 111, 112)
uncondit_formula(109,-33023,[])
uncondit_formula(109, 1,[[d, 1]])
uncondit_formula(110, 30.4375,[])
uncondit_formula(111, -10629.7210300429,[[g, 1]])
uncondit_formula(111, .321888412017167,[[d, 1],[g, 1]])
uncondit_formula(112, 9.79747854077254,[[g, 1]])
*** On date 900701 formula evaluates to 989.592054359615 ***

jed(1, 3, 2090 <Prepaid Rent >, 110).
formula(110, 113,<, 114, 115, 116)
uncondit_formula(113,-33023,[])
uncondit_formula(113, 1,[[d, 1]])
uncondit_formula(114, 30.4375,[])
uncondit_formula(115, 10629.7210300429,[[g, 1]])
uncondit_formula(115, -.321888412017167,[[d, 1],[g, 1]])
uncondit_formula(116, 9.79747854077254,[[g, 1]])
*** On date 900701 formula evaluates to -989.592054359615 ***

jed(1, 4, 2000 <Cash at Bank A >, 111).
uncondit_formula(117,-1000,[])
*** On date 900701 formula evaluates to -1000 ***

jed(1, 5, 2603 <Gearing Gain >, 112).
uncondit_formula(118, -9.79747854077254,[[g, 1]])

Exhibit 10.1, page 44
uncondit_formula(118, 1000,[[]])

*** On date 900701 formula evaluates to -4.02360515021064 ***

jeh( 2, 900601, 1,Purchase of Raw Materials)
jed( 2, 1, 3001 <Materials Price Variance >, 118).
  uncondit_formula(130, -19.177366739512,[[si, 1]])
  *** On date 900701 formula evaluates to -2164.0371403071 ***
jed( 2, 2, 2604 <CCA Gain above General Prices >, 119).
  uncondit_formula(131, -20.886026686446,[[g, 1]])
  uncondit_formula(131, 19.177366739512,[[si, 1]])
  *** On date 900701 formula evaluates to 2164.0371403071 ***
jed( 2, 3, 2021 <Polyethelene Pellets >, 120).
  uncondit_formula(132, 19.999999999999,[[si, 1]])
  *** On date 900701 formula evaluates to 22568.2242990652 ***
jed( 2, 4, 2604 <CCA Gain above General Prices >, 121).
  uncondit_formula(133, 217.8153453173,[[g, 1]])
  uncondit_formula(133, -199.999999999999,[[si, 1]])
  *** On date 900701 formula evaluates to 22568.2242990652 ***
jed( 2, 5, 3001 <Materials Price Variance >, 122).
  uncondit_formula(134, -441701698335224,[[si, 1]])
  *** On date 900701 formula evaluates to -4.02360515021064 ***
jed( 2, 6, 2604 <CCA Gain above General Prices >, 123).
  uncondit_formula(135, -48104703982985,[[g, 1]])
  uncondit_formula(135, 4.41701698335224,[[si, 1]])
  *** On date 900701 formula evaluates to 496.500934579431 ***
jed( 2, 7, 2022 <Empty Cartons >, 124).
  uncondit_formula(136, 4.39999999999995,[[si, 1]])
  *** On date 900701 formula evaluates to 496.500934579431 ***
jed( 2, 8, 2604 <CCA Gain above General Prices >, 125).
  uncondit_formula(137, 4.7919375976976,[[g, 1]])
  uncondit_formula(137, -4.39999999999995,[[si, 1]])
  *** On date 900701 formula evaluates to 496.500934579431 ***
jed( 2, 9, 2000 <Cash at Bank A >, 126).
  uncondit_formula(138, -20540,[])
  *** On date 900701 formula evaluates to -20540 ***
jed( 2, 10, 2603 <Gearing Gain >, 127).
  uncondit_formula(139, -201.240209227468,[[g, 1]])
  uncondit_formula(139, 20540,[])
  *** On date 900701 formula evaluates to -82.6448497852543 ***

jeh( 3, 900606, 2,Issue 40 x 20 kg bags of polyethelene pellets to Job 123)
jed( 3, 1, 2040 <Materials in Progress, Job 123, Step 1 >, 132).
  uncondit_formula(144, 15.9999999999998,[[si, 1]])
  *** On date 900701 formula evaluates to 1805.4579439252 ***
jed( 3, 2, 2604 <CCA Gain above General Prices >, 133).
  uncondit_formula(145, 17.4574719381906,[[g, 1]])
  uncondit_formula(145, -15.9999999999998,[[si, 1]])
  *** On date 900701 formula evaluates to -16.4554123811395 ***
jed( 3, 3, 2021 <Polyethelene Pellets >, 134).

Exhibit 10.1, page 45
uncondit_formula(146, -15.9999999999998, [[si, 1]])
*** On date 900701 formula evaluates to -1805.4579439252 ***
jed(3, 4, 2604 <CCA Gain above General Prices >, 135).
  uncondit_formula(147, -17.4574719381906, [[g, 1]])
  uncondit_formula(147, 15.9999999999998, [[si, 1]])
*** On date 900701 formula evaluates to 16.4554123811395 ***

jeh(4, 900606, 3.79 hrs factory moulding labour on Job 123)
jed(4, 1, 2041 <Labour in Progress, Job 123, Step 1 >, 137).
  uncondit_formula(149, 3.94999999999997, [[sw, 1]])
*** On date 900701 formula evaluates to 442.139560439555 ***
jed(4, 2, 2604 <CCA Gain above General Prices >, 138).
  uncondit_formula(150, 4.26520133264401, [[g, 1]])
  uncondit_formula(150, -3.94999999999997, [[sw, 1]])
*** On date 900701 formula evaluates to -5.0513034075577 ***
jed(4, 3, 3000 <Labour Clearing >, 139).
  uncondit_formula(151, -3.94999999999997, [[sw, 1]])
*** On date 900701 formula evaluates to -442.139560439555 ***
jed(4, 4, 2604 <CCA Gain above General Prices >, 140).
  uncondit_formula(152, -4.26520133264401, [[g, 1]])
  uncondit_formula(152, 3.94999999999997, [[sw, 1]])
*** On date 900701 formula evaluates to 5.0513034075577 ***

jeh(5, 900606, 4, Transfer of 10,000 1L bottles to packing)
jed(5, 1, 2045 <Materials in Progress, Job 123, Step 2 >, 143).
  uncondit_formula(155, 28.9999999999998, [[si, 1]])
*** On date 900701 formula evaluates to 3272.39252336444 ***
jed(5, 2, 2604 <CCA Gain above General Prices >, 144).
  uncondit_formula(156, 31.6416678879705, [[g, 1]])
  uncondit_formula(156, -28.9999999999998, [[si, 1]])
*** On date 900701 formula evaluates to -29.8254349408235 ***
jed(5, 3, 2040 <Materials in Progress, Job 123, Step 1 >, 145).
  uncondit_formula(157, -15.9999999999998, [[si, 1]])
*** On date 900701 formula evaluates to -1805.4579439252 ***
jed(5, 4, 2604 <CCA Gain above General Prices >, 146).
  uncondit_formula(158, -17.457797381905, [[g, 1]])
  uncondit_formula(158, 15.9999999999998, [[si, 1]])
*** On date 900701 formula evaluates to 16.4554123811467 ***
jed(5, 5, 2041 <Labour in Progress, Job 123, Step 1 >, 147).
  uncondit_formula(159, -4.99999999999994, [[si, 1]])
*** On date 900701 formula evaluates to -564.205607476626 ***
jed(5, 6, 2604 <CCA Gain above General Prices >, 148).
  uncondit_formula(160, -5.4554598068454, [[g, 1]])
  uncondit_formula(160, 4.99999999999994, [[si, 1]])
*** On date 900701 formula evaluates to 5.14231636911063 ***
jed(5, 7, 2811 <Overhead Applied - Moulding >, 149).
  uncondit_formula(161, -7.99999999999998, [[si, 1]])
*** On date 900701 formula evaluates to -902.728971962599 ***
jed(5, 8, 2604 <CCA Gain above General Prices >, 150).
uncondit_formula(162, -8.72873596909523, [g, 1])
uncondit_formula(162, 7.99999999999989, [si, 1])
*** On date 900701 formula evaluates to 8.22770619057337 ***

jeh(6, 900607, 5, Issue 825 cartons to Job 123)
  jed(6, 1, 2045 <Materials in Progress, Job 123, Step 2 >, 153).
  uncondit_formula(165, 1.81499999999998, [si, 1])
  *** On date 900701 formula evaluates to 204.806635514014 ***
  jed(6, 2, 2604 <CCA Gain above General Prices >, 154).
  uncondit_formula(166, 1.98106292755567, [g, 1])
  uncondit_formula(166, -1.81499999999998, [si, 1])
  *** On date 900701 formula evaluates to -1.79175425896119 ***
  jed(6, 3, 2022 <Empty Cartons >, 155).
  uncondit_formula(167, -1.81499999999998, [si, 1])
  *** On date 900701 formula evaluates to -204.806635514014 ***
  jed(6, 4, 2604 <CCA Gain above General Prices >, 156).
  uncondit_formula(168, 1.98106292755567, [g, 1])
  uncondit_formula(168, 1.81499999999998, [si, 1])
  *** On date 900701 formula evaluates to 1.79175425896119 ***

jeh(7, 900607, 6, 24 hrs factory packing labour on Job 123)
  jed(7, 1, 2046 <Labour in Progress, Job 123, Step 2 >, 157).
  uncondit_formula(169, .959999999999994, [sw, 1])
  *** On date 900701 formula evaluates to 107.456703296702 ***
  jed(7, 2, 2604 <CCA Gain above General Prices >, 158).
  uncondit_formula(170, 1.03708662498048, [g, 1])
  uncondit_formula(170, -.959999999999994, [sw, 1])
  *** On date 900701 formula evaluates to -1.17839426721775 ***
  jed(7, 3, 3000 <Labour Clearing >, 159).
  uncondit_formula(171, -.959999999999994, [sw, 1])
  *** On date 900701 formula evaluates to -107.456703296702 ***
  jed(7, 4, 2604 <CCA Gain above General Prices >, 160).
  uncondit_formula(172, 1.03708662498048, [g, 1])
  uncondit_formula(172, .959999999999994, [sw, 1])
  *** On date 900701 formula evaluates to 1.17839426721775 ***

jeh(8, 900607, 7, Transfer 820 cartons of 1L bottles to Finished Goods)
  jed(8, 1, 2071 <Carton 1 Litre Bottles >, 162).
  uncondit_formula(174, 31.3239999999994, [si, 1])
  *** On date 900701 formula evaluates to 3534.63528971953 ***
  jed(8, 2, 2604 <CCA Gain above General Prices >, 163).
  uncondit_formula(175, 34.1899807949053, [g, 1])
  uncondit_formula(175, -31.3239999999994, [si, 1])
  *** On date 900701 formula evaluates to -30.9228156516474 ***
  jed(8, 3, 2045 <Materials in Progress, Job 123, Step 2 >, 164).
  uncondit_formula(176, -30.3399999999992, [si, 1])
  *** On date 900701 formula evaluates to -3423.59962616811 ***
  jed(8, 4, 2604 <CCA Gain above General Prices >, 165).
  uncondit_formula(177, -33.115949984594, [g, 1])

Exhibit 10.1, page 47
uncondit_formula(177, 30.3399999999992, [[si, 1]])
*** On date 900701 formula evaluates to 29.9514183013234 ***
jed(8, 5, 2046 <Labour in Progress, Job 123, Step 2 >, 166).
uncondit_formula(178, -.655999999999985, [[si, 1]])
*** On date 900701 formula evaluates to -74.0237757009322 ***
jed(8, 6, 2604 <CCA Gain above General Prices >, 167).
uncondit_formula(179, -.716020540207442, [[g, 1]])
uncondit_formula(179, .655999999999985, [[si, 1]])
*** On date 900701 formula evaluates to .647598233541885 ***
jed(8, 7, 2821 <Overhead Applied - Packing >, 168).
uncondit_formula(180, -.327999999999992, [[si, 1]])
*** On date 900701 formula evaluates to -37.0118878504661 ***
jed(8, 8, 2604 <CCA Gain above General Prices >, 169).
uncondit_formula(181, -.358010270103721, [[g, 1]])
uncondit_formula(181, .327999999999992, [[si, 1]])
*** On date 900701 formula evaluates to .323799116770942 ***
jeh(9, 900607, 8, Return of 1 20kg bag of pellets to store)
jed(9, 1, 2040 <Materials in Progress, Job 123, Step 1 >, 172).
uncondit_formula(184, -.399999999999997, [[si, 1]])
*** On date 900701 formula evaluates to -45.1364485981303 ***
jed(9, 2, 2604 <CCA Gain above General Prices >, 173).
uncondit_formula(185, -.436597890370399, [[g, 1]])
uncondit_formula(185, .399999999999997, [[si, 1]])
*** On date 900701 formula evaluates to .39487697167192 ***
jed(9, 3, 2021 <Polyethelene Pellets >, 174).
uncondit_formula(186, .399999999999997, [[si, 1]])
*** On date 900701 formula evaluates to 45.1364485981303 ***
jed(9, 4, 2604 <CCA Gain above General Prices >, 175).
uncondit_formula(187, .436597890370399, [[g, 1]])
uncondit_formula(187, -.399999999999997, [[si, 1]])
*** On date 900701 formula evaluates to -.39487697167192 ***
jeh(10, 900607, 9, Completion of production of Job 123 - Moulding and Packing)
jed(10, 1, 2040 <Materials in Progress, Job 123, Step 1 >, 176).
uncondit_formula(188, .399999999999994, [[si, 1]])
*** On date 900701 formula evaluates to 45.1364485981237 ***
jed(10, 2, 2710 <Profit and Loss >, 177).
uncondit_formula(189, -.399999999999994, [[si, 1]])
*** On date 900701 formula evaluates to -45.1364485981237 ***
jed(10, 3, 2041 <Labour in Progress, Job 123, Step 1 >, 178).
uncondit_formula(190, -.3.9499999999999997, [[sw, 1]])
uncondit_formula(190, 4.9999999999999994, [[si, 1]])
*** On date 900701 formula evaluates to 122.066047037071 ***
jed(10, 4, 2710 <Profit and Loss >, 179).
uncondit_formula(191, 3.9499999999999997, [[sw, 1]])
uncondit_formula(191, -4.9999999999999994, [[si, 1]])
*** On date 900701 formula evaluates to -122.066047037071 ***
jed(10, 5, 2045 <Materials in Progress, Job 123, Step 2 >, 180).

Exhibit 10.1, page 48
uncondit_formula(192, -.4750000000000477, [[si, 1]])

*** On date 900701 formula evaluates to -53.599532710334 ***
jed(10, 6, 2710 <Profit and Loss >, 181).
uncondit_formula(193, .4750000000000477, [[si, 1]])

*** On date 900701 formula evaluates to 53.599532710334 ***
jed(10, 7, 2046 <Labour in Progress, Job 123, Step 2 >, 182).
uncondit_formula(194, -.959999999999994, [[sw, 1]])
uncondit_formula(194, .655999999999985, [[si, 1]])

*** On date 900701 formula evaluates to -33.4329275957698 ***
jed(10, 8, 2710 <Profit and Loss >, 183).
uncondit_formula(195, .959999999999994, [[sw, 1]])
uncondit_formula(195, -.655999999999985, [[si, 1]])

*** On date 900701 formula evaluates to 33.4329275957698 ***
jeh(11, 900606, 10, Idle time Job 123)

jed(11, 1, 2810 <Overhead Incurred - Moulding >, 185).
uncondit_formula(197, .199999999999999, [[sw, 1]])

*** On date 900701 formula evaluates to 22.386813186813 ***
jed(11, 2, 2604 <CCA Gain above General Prices >, 186).
uncondit_formula(198, .215959561146533, [[g, 1]])
uncondit_formula(198, -.199999999999999, [[sw, 1]])

*** On date 900701 formula evaluates to -22.386813186813 ***
jed(11, 3, 3000 <Labour Clearing >, 187).
uncondit_formula(199, -.199999999999999, [[sw, 1]])

*** On date 900701 formula evaluates to -22.386813186813 ***
jed(11, 4, 2604 <CCA Gain above General Prices >, 188).
uncondit_formula(200, .215959561146533, [[g, 1]])
uncondit_formula(200, -.199999999999999, [[sw, 1]])

*** On date 900701 formula evaluates to -22.386813186813 ***
jed(11, 5, 2820 <Overhead Incurred - Packing >, 189).
uncondit_formula(201, .439999999999996, [[sw, 1]])

*** On date 900701 formula evaluates to 49.2509890109882 ***
jed(11, 6, 2604 <CCA Gain above General Prices >, 190).
uncondit_formula(202, .47511034522371, [[g, 1]])
uncondit_formula(202, -.439999999999996, [[sw, 1]])

*** On date 900701 formula evaluates to -49.2509890109882 ***
jed(11, 7, 3000 <Labour Clearing >, 191).
uncondit_formula(203, .439999999999996, [[sw, 1]])

*** On date 900701 formula evaluates to -49.2509890109882 ***
jed(11, 8, 2604 <CCA Gain above General Prices >, 192).
uncondit_formula(204, -.47511034522371, [[g, 1]])
uncondit_formula(204, -.439999999999996, [[sw, 1]])

*** On date 900701 formula evaluates to 49.2509890109882 ***
jeh(12, 900606, 11, Factory wages, employees M, N, Q)

jed(12, 1, 3002 <Labour Rate Variance >, 193).
uncondit_formula(205, -.115025906735711, [[sw, 1]])

*** On date 900701 formula evaluates to -12.8753174286807 ***
jed(12, 2, 2604 <CCA Gain above General Prices >, 194).
Exhibit 10.1, page 50
The following definitions and commentary amplify the brief definitions given in
the body of the chapter. Definitions and Assumptions have been numbered to
flow on from Appendix 9.1 in the previous chapter. None of the definitions is
contentious; they are included mainly for completeness.

**Income and Income Measurement**

**Definition 15.** Net income (net loss) is the excess (deficit) of revenue over
expenses for an accounting period... [APB, 1970, para 134]

**Definition 16.** Comprehensive income is the change in equity (net assets)
of an entity during a period of transactions and other
events and circumstances from nonowner sources. It
includes all changes in equity during a period except those
resulting from investments by owners and distributions to
owners. [FASB, 1980, SFAC 3]

**Assumption 4.** Income measurement implies acceptance of some capital
maintenance concept. (The most important capital
maintenance concepts were summarized in Chapter 2.)

**Assumption 5.** There are many different acceptable income measures. A
user of accounting information may want to calculate more
than one of them.

**Comments on Income and Income Measurement**

(i) Income is the most important single measure of a firm's wealth
generation activities. Economists are interested in income because they
often assume a firm's main objective is wealth maximization. Investors
are interested in reported income because prices in stock markets around
the world seem to respond to announcements of unexpected income. In
short, income is what accountants are supposed to measure.

(ii) As discussed in Chapter 2, there are two main approaches to income
measurement. The first matches expenses against revenue (Definition
15). The second calculates the change in net assets (Definition 16).
Under the first approach there is no simple relationship between income
for a period and the change in net asset values over the period, because
some assets may be revalued without affecting income. The second
approach usually claims to be based on Hicks' definition of the income of
a person, but there is a big difference between a Hicks-based wealth
measure and what can be done in practice. To be consistent with Hicks'
approach the two wealth measures would have to be based on discounted
future cash flows for the firm as a whole. Yet, as noted above, such
wealth measures are so subjective that few accountants advocate
measuring the firm's wealth (value), and therefore income, on this basis.
For most real-world cases it seems likely that the difference between
wealth measured by discounting future cash flows, and wealth measured
by summing the values of individual assets, is likely to be large and unpredictable. So the link between Hicks' ideas and definitions like Definition 16 is tenuous.

iii) Assumption 4 recognizes that even if the wealth of a firm at two points of time were known with absolute certainty, there would still be different views about income. For instance, if a resource is purchased for $1 cash and sold one unit of time later for $2 cash in a period when the general level of prices is rising at rate $g$, and prices of the specific resource are rising at rate $s$, income under either Definition 15 or 16 may be defined as $1$ (HC), $1-g$ (CPP), and $1-s$ (CCA). Under CC/CPP an income of $1-g$ would be recognized, and this would be sub-classified into operating income, $1-s$, and non-operating holding gain, $s-g$. As if this range of income measures is not enough, if any definition of income requires the use of a price index there will also be problems in getting everyone to agree on how to measure that index.

Revenue, Expense, Gain and Loss Recognition

Definition 17. Revenues are inflows or other enhancements of assets of an entity or settlements of its liabilities (or a combination of both) during a period from delivering or producing goods, rendering services, or other activities that constitute the entity's ongoing major or central operations. [FASB, 1980, SFAC 3, para. 63]

Definition 18. Expenses are outflows or other using up of assets or incurrences of liabilities (or a combination of both) during a period from delivering or producing goods, rendering services, or carrying out other activities that constitute the entity's major or central operations. [FASB, 1980, SFAC 3, para. 65]

Definition 19. Gains are increases in equity (net assets) from peripheral or incidental transactions of an entity and from all other transactions and other events and circumstances affecting the entity during a period except those that result from revenues or investments by owners. [FASB, 1980, SFAC 3]

Definition 20. Losses are decreases in equity (net assets) from peripheral or incidental transactions of an entity and from all other transactions and other events and circumstances affecting the entity during a period except those that result from expenses or distributions to owners. [FASB, 1980, SFAC 3]

Assumption 6. The user of an accounting system may need to define which classes of exchange event generate revenue, expenses, gains and losses.

Comments on Revenue, Expense, Gain and Loss Recognition
(i) Definition 5 (in Appendix 9.1) describes an exchange as an action whereby an entity forgoes control over some resources in order to obtain control over other resources. Purchases and sales are both exchanges that involve inflows of assets or settlement of liabilities during a period. Yet, under normal accounting conventions, only sales generate revenue, not purchases. What are the rules that determine when revenue is to be recognized? According to Wolk, Francis and Tearney [1984], revenue is sometimes recognized during production, at the completion of production, at time of sale, and at time of receipt of cash; the choice presumably being made on the basis of there being a high probability of future benefits. Examples of specialized FASB standards on revenue recognition which may confound an accounting system include SFAS 45 on franchise fee revenue, SFAS 48 on revenue recognition when right of return exists, SFAS 49 on product financing arrangements, SFAS 50 on the record and music industry, SFAS 51 on cable television companies, SFAS 53 on motion pictures, and so on.

(ii) Because the rules for revenue recognition are so hard to define, Assumption 6 recognizes that it may not always be possible for the Interpreter to distinguish revenue-earning events from all others. The accountant controlling the system may have to define which types of events are to be used for revenue recognition. The same comments apply to recognition of expenses, gains, and losses. However, once the accounting treatment for a given event type has been defined, the Interpreter can process all future events of that type without human intervention.
Chapter 11. Summary and Conclusions

Yesterday, US President George Bush announced a cease-fire in the US-led Allied war against Iraqi president, Saddam Hussein. In the past months, Iraqi electricity and water utilities, communications facilities, and munitions factories were devastated, apparently with unprecedented accuracy, by "smart bombs" and equally smart missiles. According to today's newspaper reports, over 100,000 Iraqis have been killed, yet Allied casualties during the entire period numbered fewer than 200. Media reports say that "high technology" military equipment, much of it computer controlled, has completely changed the rules of war.

If information-based technology can have such a dramatic impact on one aspect of human endeavour, one wonders if it might not also have a major impact on other information processing activities. Perhaps the technology for accounting record-keeping has changed so much since Pacioli's days in Venice that the whole process of doing accounting should be reconsidered from first principles. Designers of computer-based accounting systems in the past thirty years have tended to focus on producing existing information cheaper, or faster, but few questions have been asked about whether better accounting information could be produced by using a computer. It may be timely, therefore, that this thesis asks whether cost-effective, computer-based accounting systems can be used to generate better accounting information than existing transaction processing accounting systems.
This question is much broader in scope than is typically chosen for a doctoral thesis. As a consequence, the conventional hypothesis-testing thesis structure is not appropriate. Instead, Part A of the thesis, Chapters 2 through 5, was devoted to gathering and summarizing information about what might constitute "better" accounting information; Part B, Chapter 6, proposed an architecture for computer-based accounting systems that might be used to produce that "better" accounting information; and Part C, Chapters 7 through 10, "fleshed out" the skeletal architecture provided in Chapter 6, providing details of Formula Accounting (FA), the Resources and Exchange Events (REE) data model, a possible way to build an Interpreter program to generate FA journal entries, and an example of Multiview Accounting "in action".

11.1 Detailed Chapter Summary

11.1.1 Part A: Requirements Determination

The problem addressed in Part A of the thesis was to decide what computer-based accounting might be asked to do "better" in future. In a conventional systems design exercise, requirements determination usually involves (a) documenting existing systems [DeMarco, 1978], (b) determining what the future system should do by asking users about their information requirements [Davis, 1982] and/or attempting to deduce information requirements from high-level goals such as managers' critical success factors [Rockart, 1983]. A roughly similar approach to requirements determination was used in this thesis. The major difference being that the system design contemplated in this thesis is
intended for use in many organizations, not just one, so the organizational issues that pervade many conventional systems design exercises were not an issue in this thesis.

(a) Documentation of existing systems

Because of the lack of published information on computer-based accounting, "documentation of existing systems" was achieved by analyzing 224 responses from two mail questionnaires sent to a total of 400 randomly selected large Australian firms, and by personal reviews of widely-used packaged accounting software. The information from these various sources, presented in Chapter 4, showed that computer-based transaction processing accounting systems (TPAS) are in wide-spread use throughout Australia (and presumably the Western world); that they differ widely from industry to industry (e.g., banking vs. manufacturing); and that their databases contain quite comprehensive descriptions of the activities undertaken by the different firms. In addition, it was found that in many cases these special-purpose TPAS were being used as sources of journal entries for the firms' general ledgers. The main objective of the designers of these systems seemed to be to assist managements by making it cheaper, faster, and easier to record details of transactions, and by facilitating reporting of the firms' states of affairs at any given time. In the minds of the accountants responsible for running these TPAS, a system was useful if the information it contained was complete, accurate, timely, and well-formatted. Ease-of-use was a lesser consideration, though still significant.
Requirements for "better" accounting systems\(^1\) were determined by reviewing the accounting literature from the last 80 years. The purpose of this review was not to try to summarize all that had been said, but rather, to extract from the literature the principal recommendations for improvements to accounting.

The review commenced in Chapter 2, where a top-down, almost deductive approach was adopted. To begin with, three main "Objectives of Accounting" were identified. These were:

1. To provide management with information useful for planning and control.

2. To provide external users with information useful for predicting the future cash flows and the earning power of a firm.

3. To provide accountability information useful for contract monitoring.

Then, by considering each of these objectives in some depth, it was decided, first, that the most fruitful place for widely generalizable research into accounting information for management planning and control was at the general ledger level; second, that the most useful form of information that an accounting system can reasonably be asked to produce for predicting future cash flows and the earning power of a firm appeared to be an income statement.

---

\(^1\) This is a far more fundamental question than "requirements for better Computer-based accounting systems".
supported by a balance sheet and details of past cash flows; and third, that a
detailed historical record of all transactions is required for accountability
reporting. These requirements are compatible with one another, so it is
reasonable that future computer-based accounting systems should be asked to
meet all three requirements.

Now, there is little uncertainty about which details of transactions should be
stored for accountability reporting. Nor is there much uncertainty about how
cash flows should be measured and reported. But income measurement is quite
another story. Almost all the (vast) normative accounting literature of the
period 1955-1970 was aimed at the closely-related questions of asset valuation
and income determination. Ignoring Present Value Accounting because of the
problems of estimating future cash flows, the five major systems of asset
valuation and income determination advocated in the normative literature, i.e.,
HC/GAAP, CPP, CCA, CoCoA, and CC/CPP, were summarized in Section 2.4.
Apart from HC/GAAP the other four methods of accounting all attempt to
incorporate into the accounts the effects of changing prices.

To assess the merit of some of the variants of CPP, CCA, and CC/CPP
accounting, the empirical literature on the inflation accounting experiments in
the US (SFAS 33 [FASB, 1979]) and UK (SSAP 16 [ASC, 1980]) was reviewed in
Chapter 3. There are two potential problems with these studies. First is the
data clustering problem. Data for the studies all come from a period when
general inflation (as measured by the CPI) in both countries was less than 20% per annum. Second, the most significant empirical studies all rely on the
assumption of semi-strong efficiency in the US and UK capital markets. This
casts some doubts on their validity because, as was shown by Foster [1979, 1986] and Ou and Penman [1989], there are clearly some occasions where market prices have not reflected all publicly-available information. Thus the conclusions of the market-based studies of inflation accounting reports should not be viewed as tests of the information content of accounting reports in any absolute sense. They do, however, give an indication of the extent to which market analysts used the information reflected or conveyed in CCA reports to value securities during a period of 5% to 20% inflation.

The studies indicate that during the test periods in the US and the UK, market analysts found the information content of HC/GAAP and Current Cost reports to be very similar. In the US, where LIFO is the normal inventory flow assumption\(^2\), correlations between the two measures during 1980 and 1981 were around 0.7, i.e., both measures of income contained similar information. In the UK, Carsberg [1984a] found so many problems with physical capital maintenance for companies in the extractive, shipping, and construction industries that he recommended exempting companies for whom maintenance of operating capacity was not relevant from CCA reporting. To overcome this problem, he toyed with recommending financial capital maintenance, i.e., CC/CPP accounting, for future inflation accounting standards, but decided to stick with CCA for most firms on the grounds that investment analysts found SSAP16 current-cost dividend cover a useful number. In the US, the Appendix to SFAS 89 [FASB, 1986] comes down more strongly in favour of financial capital

\(^2\) Use of LIFO for inventory valuation means that the major source of difference between GAAP and CCA income is due to the difference between depreciation based on historical cost and what might be loosely termed "replacement cost".
maintenance. Since professional practice in both countries seemed (before their inflation accounting experiments were abandoned) to be converging on disclosure of CC/CPP accounting numbers\(^3\), the illustrative examples of inflation accounting presented in Chapters 7 through 10 of this thesis all use CC/CPP accounting.

In addition to the deductive/normative approach to Requirements Determination employed in Chapter 2, a second source of ideas on how computer-based accounting systems should be designed is the academic literature on Accounting Information Systems. This literature, reviewed in Chapter 5, is disappointing because so few people have contributed\(^4\). By stretching the definition of computer-based accounting to include Multidimensional accounting, the major contributors have been Ijiri [1966, 1975], McCarthy [1978, 1979, 1982], and Weber [1977, 1986].

This thesis was heavily influenced by all three authors. First, whilst not directly affecting the requirements for computer-based accounting systems that surfaced as Design Principles in Chapter 6, Ijiri's work on accountability reporting, Multidimensional accounting, and causal double-entry are at the very core of the proposals presented in this thesis. Second, McCarthy's REA model [1982], his suggestion that conventional accounting reports should be defined as

---

\(^3\) It seems likely that future standards would emphasize disclosure, rather than any single profit measure. This would require disclosure of information useful for calculating income using both maintenance of financial capital and operating capability.

\(^4\) It is not clear why the literature is so sparse. It may be that computer-based accounting is not a "hot" topic among journal editors, and many manuscripts have been rejected. Alternatively, it may be simply lack of interest.
views of an underlying set of facts about the enterprise, and his preparedness to abandon double-entry accounting were also major factors that shaped the Resource and Exchange Events (REE) model and the Interpreter in this thesis.

Third, Weber's [1986] use of packaged software as a source of evidence about how computer-based accounting is practiced today was the stimulus for most of the work reported in Chapter 4.

Thus, by the end of Part A, many suggestions for improved accounting systems had been considered, but no firm decisions had been made about the directions that this study should go. The stage was set for Part B.

11.1.2 Part B: An Architecture for Computer-based Accounting

On the basis of the evidence presented in Part A, together with additional evidence that had to be collected to resolve uncertainty about the merit or otherwise of inflation accounting, two fundamental Design Principles were enunciated in Chapter 6. The first was that changes in the prices/values of resources owned or controlled by an organization are so fundamental to the measurement of economic activity that they should, if possible, be reflected in accounting reports. The second was that although it is unrealistic to attempt to specify general principles for the design of transaction processing systems for different organizations, it may be confidently predicted that each firm's special-purpose transaction processing systems will capture the key elements of almost all economic events undertaken by the firm. It is therefore reasonable to treat the data from these heterogeneous special-purpose systems as the prime source of information for the generation of accounting reports.
In short, by the end of Section 6.2, Chapter 1's broad requirement for "better" accounting information had been sharpened considerably. The design goals for the thesis were now to

(a) concentrate on providing information for management planning and control at the general ledger level;

(b) seek to create a system of accounting capable of coping with changing prices, for both internal and external reporting purposes, particularly for income measurement (Design Principle 1); and

(c) rely on the existence of special-purpose feeder systems as the prime source of data for the general ledger system and for accountability information (Design Principle 2).

Looking back, these decisions were present in embryonic form in the Objectives of Accounting identified in Chapter 2, but there was much weighing up of the pros and cons of the various proposals before they could be accepted as design goals for the thesis. Moreover, it was not until the practical computing systems described in Chapter 4 had been surveyed and reviewed in depth that it was really clear that they could, and should, be relied upon as an almost complete source of Sorter's "Events" data [1969] and McCarthy's REA data [1982] for the accounting systems of the future.
The most contentious proposal in this thesis is the decision to build inflation tolerance into the accounting system\textsuperscript{5}. Not surprisingly, it was also the most difficult to justify. On the one hand, almost all the normative literature of the period 1955-1970 is strongly of the opinion that historical cost accounting produces misleading information in times of rapidly changing prices; it simply does not make sense to add dollars of different purchasing power. On the other hand, the CCA experiments of the early 1980s (SFAS 33 and SSAP 16) had badly shaken the accounting profession's confidence in its ability to produce meaningful inflation accounting numbers.

Resolution of this conflict was achieved by collecting additional empirical evidence that showed that during periods of very high inflation in various countries around the world, historical-cost accounting had, as the normative theorists either knew or predicted, become entirely unworkable. It was concluded, therefore, that the benefits of inflation accounting must only become apparent when general price-level changes approach the levels found in Brazil in the 1970s, e.g., when they exceed, say, 15\%-20\% p.a.. Although it would not be apparent in market-wide studies, managers and investors might also find inflation accounting information useful for firms whose major assets or liabilities experienced specific price changes of this order, even at lower levels of general inflation.

\textsuperscript{5} As discussed in Chapter 1, the terms "inflation tolerance" and "inflation accounting" are used in this thesis as generic terms for any sort of accounting that attempts to deal with changing prices.
Of the systems of accounting reviewed in Chapter 2, the most promising seems to be Current Cost/Constant Purchasing Power (CC/CPP) accounting. In CC/CPP accounting, the use of deprival values, not just replacement costs, eliminates some of the downright silly asset values that CCA produces. In addition, the use of financial capital maintenance, not maintenance of operating capacity, overcomes problems where wealth is clearly increasing yet CCA reports no gains. But many problems, e.g., the appropriateness of price index adjustments, technological obsolescence, not to mention the scope for "creative accounting", still remain. As was shown in Chapter 2, Section 2.3, income measurement is inherently subject to large measurement error, and any move from Historical Cost forces accountants to consider problems that may otherwise be conveniently ignored. In the author's opinion, a relatively simple system like that outlined at the end of Section 6.1, is probably the most practical compromise between desires for technical precision and the need for rigid rules to thwart creative accounting. However, the architecture proposed in this thesis is not restricted to such systems.

The decision to make future accounting general ledger systems both inflation-tolerant, and based on facts drawn from the corporate database, determined the basic architecture for those future systems, almost by default. That architecture, introduced in Section 6.4 and reproduced on the next page as Figure 11.1, relies on a Formula Accounting General Ledger system, fed with special information on current prices, to build inflation-tolerance into the basic fabric of the general ledger and profit measurement system.
The architecture depicted in Figure 11.1 provides two paths from the host organization's subsystems to the FA general ledger. In the upper path, the subsystems themselves generate FA journal entries (which is why they need access to the special information, such as price index series, shown in Figure 11.1). In the lower path data are presumed to be available in a pre-defined, standard format (e.g., the REE data model), and a program called an Interpreter generates the journal entries. Either or both paths might be used, but investigating the lower path enables the data requirements for either path to be identified explicitly. The lower path was therefore investigated in Part C of the thesis.

11.1.3 Part C: Details of the Proposed Architecture

Formula Accounting is a simple technique, entirely dependent on the power of spreadsheet-like computer technology, that enables ledger balances to be
programmed to change automatically over time. For almost all of the 500 years since Pacioli [1496] it has been impractical for accountants to do more than record and classify details of economic events, then compute summary statements of economic position. However, as suggested in the opening paragraphs of this chapter, computers have reduced the effort required to record, classify, and prepare reports so much that accountants can now afford to expand their horizons. Formula Accounting is one possible way to use computer power to add value to the accounting function.

In a Formula Accounting general ledger system, constant-value journal entries and constant-value account balances are replaced by formulae like those in spreadsheets. Formulae can be defined to depend on any sort of variable that is likely to change in value over time. Variables illustrated in Chapter 7 included time itself (depreciation), current market prices (including prices of shares, foreign exchange, commodities, and fixed-interest securities, all assumed to be available electronically), other ledger account balances (conceptually similar to references to other cells in a spreadsheet), and general and specific price indices (for inflation accounting). For direct entry of journal entries from a keyboard, it was suggested that standard formula templates could be set up, so that the operator only needed to "fill in the blanks" on a screen. It was also suggested that it would be helpful if every journal entry had its own local spreadsheet readily accessible, e.g., accessed by pressing a function key. This would enable complex journal-entry formulae to be defined very simply.

The major technical difficulty with Formula Accounting occurs when complex, conditional formulae are posted to already complex, conditional formulae in
ledger account balances. To speed evaluation of ledger account balances it is desirable to simplify an account balance formula as much as possible as each new journal entry is posted to it. Techniques exist for symbolic computation, e.g., in software products such as Mathematica and Maple, but pursuit of a general solution to the problem of representing and simplifying formulae for Formula Accounting is beyond the scope of this thesis.

To provide the reader with a clearer sense of the target FA journal entries an Interpreter program in the proposed architecture would be required to produce, a prototype Formula Accounting system was described in Chapter 8. The data structure chosen in Chapter 8 is sufficient for simple conditional formulae like those required for depreciation. However, it is not intended to do more than give an indication of the difficulties of the general case of posting (adding) formulae of arbitrary complexity.

When using a Formula Accounting system, procedures such as post_je (post journal entry), list_je (list journal entry), and tb (prepare "trial balance"), defined in Table 8.2, are conceptually no different to their conventional accounting counterparts. A trap for users accustomed to conventional ledger systems is the need to distinguish between closing an account and transferring its balance to another account. For instance, the balance of a Depreciation Expense account is transferred to Profit and Loss (or Manufacturing Overhead) at the end of each period, leaving a formula in the Depreciation Expense account that evaluates to zero on that day. However, when the asset is sold, the Depreciation Expense account for the asset is closed, and the formula in the account is erased. Apart from this, Formula Accounting is a straight-forward
extension to conventional double-entry accounting that gives a vital new meaning to Chamber's concept of "continuously contemporary accounting".

Chapters 9 and 10 considered the problems of generating FA journal entries automatically. The first step in that process was to define the interface between the subsystem databases and the program that was to generate the FA journal entries. As both Ijiri [1975] and McCarthy [1979, 1982] had previously observed, conventional accounting systems use journal entries to record (the financial implications of) a firm's exchange events. This involves setting up ledger accounts for different classes of resources. However, McCarthy [1982] decided that accounts should not be used as a fundamental data modelling construct:

It is the primary contention of this paper that the semantic modeling of accounting object systems should not include elements of double-entry bookkeeping such as debits, credits, and accounts. [pp.559-560]

This, in turn, led him to his problematical "duality" construct discussed in Chapter 5. However, as accountants have known for centuries, an account is a convenient way to model stocks of resources, including those controlled by different parties. Accounts also decouple events, so that there is no need for duality relationships linking, say, a series of sale of inventory events in a retail store to the purchase of that inventory\(^6\). Thus for the REE model presented in

\(^6\) If duality relationships were captured in conventional accounting systems today, there would be no need for LIFO/FIFO flow assumptions, since each sale of inventory would, by definition, be linked to the purchase event that acquired the inventory. It is the difficulty in connecting these two types of event that is the problem with McCarthy's "duality".
Chapter 9 the notion of an account (identical to Ijiri's Multidimensional account) was retained as a fundamental modelling construct. Exchange Events could then be recorded simply as increases and decreases in quantities of resources held in different Asset and Obligation accounts. Exchange Events are a powerful, compact way of describing economic events.

Now, a history of Exchange Events should be quite sufficient for stewardship or accountability reporting, but those data must be transformed into journal entries by an accountant (using GAAP or their alternatives) before income statements and balance sheets can be prepared. In other words, if Exchange Events are the raw material for accounting ledger systems, and income statements and balance sheets are the finished product, journal entries must be regarded as an intermediate product. The reason is that journal entries embody details of both the event, and the accountant's interpretation of that event. To prepare a journal entry the accountant must ask questions like: Is this a revenue-recognition event? Is this an asset than needs to be depreciated, and if so, what is the asset's expected life? Answers to questions like these are embodied in every journal entry.

Thus, in addition to its Exchange Events, the REE model must also contain sufficient information about resources, and the way their values are expected to change over time, for FA journal entries to be generated. In REE, resources are classified into (i) those whose (physical) form does not change over time but

---

7 Formula Accounting journal entries do this even better than conventional constant-value journal entries because they embody the accountant's expectations about future changes in value as well.
whose value may change, e.g., shares, (ii) those whose (physical) form and value both change but where the resource is considered to be of the same type, e.g., a truck, and (iii) those that change so much that they are treated as different resources at different points in time, e.g., for a forestry company, seedlings grow to become trees. Suggestions are also made in Chapter 9 for describing resources that are parts of some composite resource, fixed-interest securities as time-composite resources, and resources in manufacturing processes. The guiding principle in defining the REE data model was that the data must be necessary for the generation of FA journal entries. Chapter 9 focused on some example trading and manufacturing Exchange Events; additional information might be needed in other firms or industries.

In Chapter 10, an effort was made to see if the rules used by accountants in the preparation of journal entries could be encoded in a computer program. Because the "value added" by that program is an accountant's interpretation of each Exchange Event, the program was called an Interpreter. Of course, as was shown in Chapter 4, many Accounts Payable, Accounts Receivable, and even Fixed Asset systems today already generate journal entries that feed straight into a general ledger. So the idea of automating the generation of journal entries is nothing new. Generation of journal entries by such subsystems is shown in the upper path in Figure 11.1. Because the subsystems "know their own data", this might well prove the most cost-effective way to generate Formula Accounting journal entries in future.

However, the Interpreter envisaged in this thesis (shown on the lower path in Figure 11.1) offers the possibility of more centralized control over accounting
rules. Such centralized control of accounting policy might be attractive to many organizations. For instance, a professional accounting firm might find it could offer cheaper audits if its client used the accounting firm's own "certified" Interpreter (probably tailored for each particular client) to maintain its accounts. An Interpreter that generated almost all journal entries, and simultaneously logged all suspicious or unusual patterns of transactions would be a powerful agent for accounting control in the client's organization.

Another possible benefit of the REE-Interpreter approach for generating FA journal entries is its capacity to support Multiview Accounting. In present-day accounting systems there is considerable human input into the process of collecting data and preparing accounting reports. This is expensive, and it restricts the range of reports it is feasible to prepare. However, with an automated system, where the rules of accounting are encoded into the system, it would seem feasible to maintain, say, an HC/GAAP set of books, a CC/CPP set of books, and a tax set of books. This is more-or-less what the ASOBAT report recommended [AAA, 1966] and, in a time of high inflation, the combination of reports might meet many information needs. A more immediate application of Multiview Accounting might be for a multinational firm which needs to report under different accounting standards in different countries. For instance, Australian Accounting Standards forbid Australian firms to use LIFO for inventory valuation, whereas most US firms do use LIFO. Australian subsidiaries of US corporations must therefore maintain two sets of inventory records. Rules for depreciation and lease capitalization are also different in the two countries. If there were enough differences of this sort, a firm obliged to
report under both sets of rules might find it worth-while to keep two full sets of books.

For both the above reasons, i.e., centralization of Interpretation, and Multiview Accounting, it seemed worth investigating, in this thesis, the feasibility of building an Interpreter. The prototype Interpreter was constructed using a CASE statement (based on Event Type) as shown in Figure 10.4, and a series of core modules that could be used by all event-handlers. Its basic structure seems sound, although it may be over-dependent on the Event Type for deciding how to process Exchange Events. It also seems desirable that the core modules should be as general as possible so that FA journal entries can be generated when "anything is exchanged for anything" not just, say, where Finished Goods inventory is exchanged for Accounts Receivable, as is the case in most of today's Accounts Receivable systems.

What is also clear is that when the Interpreter is called upon to do more than just simple trading accounting journal entries, it starts to need more detailed information about resources than might initially have been thought necessary. Thus, in addition to obvious requirements for expected-value data for depreciating or amortizing different classes of assets, the prototype Interpreter needed considerable extra information for accounting for a manufacturing firm. When coding the Manufacturing Event-handler, all the problems of job costs,

---

8 For instance, one could contemplate a system where all resources were classified as of type "revenue generating", or not. Exchange events resulting in decreases of revenue generating resources would normally be treated as revenue vents. Alternatively, a scheme like this could be used to check the validity of Event type classifications.
process costs, standard costs, joint costs, set-up costs, overhead costs, cost allocation, by-products, treatment of scrap and rejects, and even the meaning of Exchange Events in this environment had to be considered anew. These problems have all been thoroughly described and discussed in management accounting textbooks and literature, and it is really a fairly mechanical process to define the necessary facts in the REE database and encode the appropriate rules in the Interpreter. However, the solutions offered in Chapter 10 are more organization-specific than one might have hoped for.

11.2 This Thesis's Contributions to Knowledge

The first original contribution of this thesis is Formula Accounting. Formula Accounting (FA) is attractive because it is consistent with all the principles of double-entry accounting, yet it also provides a simple mechanism to enable account balances to change automatically over time. Formula Accounting makes it relatively simple for ledger account balances to be continuously expressed in current values, which is ideal for inflation accounting. Formula Accounting may be used for general price-level adjustments, specific price adjustments (replacement cost, deprival value, or exit-prices), or simply for the allocation of historical costs over time (see the depreciation example in Figure 7.1).

The second original contribution of this thesis is the architecture depicted in Figure 11.1, and the possibilities it offers for Multiview Accounting. Figure 11.1 shows two pathways for data to flow from the transaction processing subsystems (on the left) to the FA general ledger (on the right). Both
are feasible, and both could be used in conjunction with one another. If the upper pathway is used, the subsystems generate FA journal entries themselves. If the lower pathway is used, the subsystem databases must be so structured that an REE view (or something similar) may be defined. As envisaged in this thesis, a full Multiview Accounting system would consist of a number of Formula Accounting general ledgers, and the interface programs on the left (using the upper or lower paths, or both) would be capable of generating FA journal entries according to different sets of accounting rules.

How valuable Multiview Accounting would be to either management or external users is unclear. For instance, an investor would probably be more interested in a glimpse at management's expectations for the future (if such expectations could be trusted) than two or more different views of the past. However, there do appear to be situations, e.g., the multinational firm today, or any firm in a country experiencing inflation of 20% p.a. or more, when the ability to produce different accounting interpretations of the one underlying "reality" might be useful. It is also considered important that the subsystem databases (on the left of Figure 11.1) should be retained for ad hoc analyses by management.

Development of the REE model, which is an extension of both Ijiri's Multidimensional Accounting [1975] and McCarthy's REA accounting model [1982], is the third original contribution of this thesis. Where Formula Accounting extends double-entry accounting in one direction, by replacing constant dollar figures by formulae, the REE model extends double-entry accounting in another direction, by maintaining accounts in terms of units of different resources. The Exchange Events of the REE model are like journal
entries, but they describe the (physical) resources exchanged, not their money value. Deferring valuation to a later step in the accounting process opens up the possibility of using different valuation rules to generate different accounting views of the same underlying economic events, i.e., of Multiview Accounting.

The fourth, original contribution claimed for this thesis is the concept of an Interpreter program, and the demonstration of its feasibility. As shown in Chapter 10, once an REE view of the data in the subsystems is available, it is not too difficult to write an Interpreter program to generate FA journal entries. It is envisaged that Interpreter programs would be need to be tailored to different industries. They would also need to be maintained over time, as the organization and accounting standards changed.

The four ideas listed above are all from the second half of the thesis. The following three contributions from the first half of the thesis also seem worth noting:

(a) The approach to measurement embodied in the "better to disclose details" rule, and second, the notion that good accounting systems have lower measurement error than other systems. These ideas were discussed in Chapter 2. Others have discussed aggregation rules, e.g., MacNeal [1939, p.171], Miller [1973], but no one seems to have suggested a simple idea like the "better to disclose details" rule for reporting asset and liability values. Others have also considered measurement error [Jaedicke, Ijiri, and Nielsen, 1966], but no one seems to have made the
simple observation about the effect of independent valuation at start and end of period on income measurement. It was argued in Chapter 2 that historical cost "matching" became accepted practice because it has lower measurement error than the alternatives. Low measurement error is desirable because there are so many different parties, most with different competing interests, that accounting information is intended to serve.

(b) Much of the material in Chapter 4, on computer-based accounting systems in practice today, was collected specially for this thesis. It would be useful material to include in an Accounting Information Systems text.

(c) Finally, the detailed analysis of the arguments for and against inflation accounting that identifies the data clustering problem with the empirical studies reviewed in Chapter 3, and culminates in the simple conclusion that there must be some cut-over point where inflation accounting \(^9\) becomes necessary, could be regarded as an original contribution to knowledge. It seems a pity that all the energy devoted to normative accounting research in the 1960s and 70s tends to be dismissed today as irrelevant. It is not. The seemingly trivial conclusion that there must be some cut-over point when inflation accounting becomes necessary, and

---

\(^9\) This term is used in the sense indicated in Chapter 1, i.e., any system of accounting that attempts to deal with the problems of general changes in purchasing power of the unit of account and/or specific changes in entry or exit prices.
that this is probably around the 15% to 20% p.a. mark, helps to place both the normative and empirical work in perspective.

Having identified the thesis's major contributions to accounting knowledge, it also seems appropriate to ask if the stated objective of the thesis, i.e.,

whether cost-effective, computer-based accounting systems can be used to generate better accounting information than existing transaction processing accounting systems

has been achieved. The honest answer is that one cannot say. The questions raised in this thesis are too general for a simple yes/no answer.

It seems likely that a cost-effective Formula Accounting system could be built. It also seems likely that it would be cost-effective to generate FA journal entries automatically from subsystem data. It is not clear, however, that the information produced by such a system would be any better than that available at present. If it is accepted that accounting systems should be inflation-tolerant, or more generally, capable of coping with changing prices, one could probably build a fairly strong case to argue that the computer-based support offered by the architecture proposed in this thesis is capable of providing "better" accounting information more elegantly, and cost-effectively, than existing transaction processing accounting systems. Moreover, since it seems unlikely that periods of high inflation or rapid priceChanges of major assets will never occur again, anywhere in the world, there seems a good chance that a system designed along the lines proposed in this thesis could actually deliver "better" information.
11.3 **Suggestions for Future Research**

The acid test of the architecture proposed in Figure 11.1 would be to build a full-scale working system, not just a prototype. Development could be proceed on a number of fronts simultaneously:

1. It is envisaged that greatest practical difficulties would be encountered in trying to define an REE database as a view of the subsystem databases. It would therefore be worthwhile to try to define an REE view of either all the data for a particular firm or a particular suite of software packages. Perhaps the safest short-term solution would be to create real tables (e.g., using SQL command CREATE TABLE, not CREATE VIEW) and write interface programs to update those tables with data from the subsystems. Attempting to create such a database would probably unearth all sorts of problems. However, if the reasoning in Section 6.2 is correct, and there seems every reason to believe it is, all, or almost all, the Resource-Event-Agent Exchange Event data required for income measurement will be available in the subsystem databases.

2. Another direction for research would be to focus on just one firm, e.g., in manufacturing, mining, or banking, to see (i) if some alternative or improvement to the REE data model could be developed, or (ii) how difficult it would be to build an Interpreter that successfully generated FA journal entries for all that firm's interests and activities.

3. Formula Accounting is still a very new idea, and there are many implementation details to be worked out. The prototype system described in
Chapter 8 indicates that the idea is feasible, but it does nothing to exploit the idea of a local spreadsheet for each journal entry and, as a consequence, sidesteps the problem of updating complex conditional ledger account balances with similarly complex journal entries.

What the writer would like to do, would be to take a general ledger package from a good suite of accounting software packages, a good spreadsheet program, and a good symbolic computation package, and marry the three together into one software product for Formula Accounting.

At the human interface (for manual entry and review of Formula Accounting journal entries), it would be useful to have some standard way (i) for representing formulae (both journal entry and ledger balance formulae) on a screen, and (ii) for declaring references to external variables, e.g., price index series and prices quoted on the Stock Exchange. (References to such variables could, presumably, use something like Lotus 123's notation for reference to cells in other spreadsheets.) Rules for storing journal entry templates, and for checking to ensure that journal entries balance over time are also required. The problems illustrated in the bond-valuation example in Chapter 7 give a hint of the interesting difficulties that might need to be solved.

4. The obvious next step after Formula Accounting was working would be to use either the upper or lower path of Figure 11.1 (whichever seemed easier) to build a Multiview Accounting system to link the other packages in the suite to the new Formula Accounting system.
The above four points all relate to extending, developing, and testing the architecture shown in Figure 11.1. On quite a different tack, it was suggested in Chapter 7 that novice students might find it easier to learn Formula Accounting than conventional accounting. It would be interesting to know if this claimed advantage of Formula Accounting, i.e., that it is easier for novices to learn, is real.

(6,700 words)
Appendices: Prolog code for the prototype Interpreter
Appendices

Details of the six Appendices, each containing Prolog code, are given below (this information was also given in Chapter 10). Each appendix is numbered separately, with an abbreviated name before the page number. There is an index to all procedures at the end of each appendix.

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Lines</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Formula</td>
<td>926</td>
<td>Contains almost all the code for the Formula Accounting system. This code performs the listing, posting, balance-checking, account-closing, and formula-evaluation functions described in Chapter 8 and illustrated in Exhibit 8.1.</td>
</tr>
<tr>
<td>2. Interpreter</td>
<td>1391</td>
<td>Contains the bulk of the code for the Interpreter. This code initializes the Multiview system, performs causal double-entry reasoning, generates Formula Accounting journal entries (including those for depreciation, amortization, and standard costing variances), maintains inventory records (LIFO, FIFO, and Standard Cost), and contains all the code for the three different variants of inflation accounting. As shown in Interpreter 1, it uses mapping tables like those shown in Figure 10.3 to link together the REE and conventional accounting models.</td>
</tr>
<tr>
<td>3. Special Event Handler</td>
<td>504</td>
<td>This code contains nine specific Event handlers for the Interpreter. These are as follows: cash_payment, cash_receipt, credit_sale, donation, issue_shares, manuf_event, open_accounts, production_summary, and sale_depreciable_asset. Some of these modules are quite short, e.g., cash_payment is 15 line long, because they use code in the &quot;core&quot; Interpreter wherever possible.</td>
</tr>
<tr>
<td>4. Prolog Library</td>
<td>350</td>
<td>Low-level Prolog procedures like once, sum, max, write_fixed_length, is_a, and date_function, that extend the minimal commands provided by basic &quot;Clocksin and Mellish&quot; Prolog, e.g., there is not easy way to format printed output in Prolog. (Prolog is, however, excellent for manipulating &quot;tables&quot; of data.)</td>
</tr>
<tr>
<td>5. Trading Database</td>
<td>929</td>
<td>Trading database, containing all database &quot;tables&quot; including the facts for Table 9.1 in lines 616-642. This database was used in Runs 1 and 2 in Exhibit 10.1. Tables are organized in three main groups, being those for Formula Accounting, the REE model (including the Exchange Events), and the Interpreter, respectively.</td>
</tr>
<tr>
<td>Appendix</td>
<td>Lines</td>
<td>Contents</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>6. Manufacturing</td>
<td>873</td>
<td>Manufacturing database, containing all database &quot;tables&quot; including the facts for Table 9.2 in lines 572-625. When used in Runs 3 and 4 of Exhibit 10.1, this database completely replaced the Trading database. Many of the tables in the two databases, e.g., the various object-type hierarchies, are similar.</td>
</tr>
</tbody>
</table>
APPENDIX 1: FORMULA ACCOUNTING

This Appendix contains 926 lines (19 pages) of Prolog code for Formula Accounting. The procedures listed in Chapter 8, Table 8.2, may be found on the following pages:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_je(JE_no,Date)</td>
<td>2</td>
</tr>
<tr>
<td>list_jes(Date)</td>
<td>2</td>
</tr>
<tr>
<td>list_bal(GL_ACno,Date)</td>
<td>6</td>
</tr>
<tr>
<td>tb(Date)</td>
<td>5</td>
</tr>
<tr>
<td>money_tb(Date)</td>
<td>6</td>
</tr>
<tr>
<td>check_balance(JEno)</td>
<td>9</td>
</tr>
<tr>
<td>post_je(JEno)</td>
<td>12</td>
</tr>
<tr>
<td>close_revenue_acs(Date)</td>
<td>14</td>
</tr>
</tbody>
</table>

There is an index to all procedures in this Appendix on pages 21-22.
/* LIST JOURNAL ENTRY */

list_jes:-
report_date(Report_date),
list_jes(Report_date).

list_jes(Report_date):-
once(
  (start_accounting_period(Start_accounting_period);true),
nl,write('Journal entries for REEFA LTD, for period from '),
write(Start_accounting_period),write( ' to Report date '),write(Report_date),nl,
price_level_accounting_switch(S1),
fifo_lifo_switch(S2),
/* leased_assets_on_off_balance_sheet(S3), */
write('Journal entries for '),write(S1),write(', '),write(S2),write(', and leased assets '), */
write('balance sheet.'),nl,nl
write('for '),write(S1),write(', '),write(S2),nl,nl
).
write(jeh(JEno,Date,Eno,Narrative)),
one(list_je(JEno,Report_date)),
fail.

list_je(JEno):-
report_date(Report_date),
list_je(JEno,Report_date).

list_je(JEno,Report_date):-
writeln(jeh(JEno,Date,Eno,Narrative)),
write(jeh(JEno,Date,Eno,Narrative)),nl,
list_jeds(JEno,Report_date),nl.

list_jeds(JEno,Report_date):-
jed(JEno,L,GL_ACno,Fno),
one(
  tab(3),write(jed('),write(JEno),write(', '),
write(L),write(', '),
write(GL_ACno),write(' '),
chart_of_accounts(GL_ACno,Description,Class),
write('<'),write fixed_length(Description,45),write('>'),
write(Fno),write(').'),nl,
list_formula(Fno,Report_date,_)
),
fail.

list_jeds(JEno,Report_date).

/* LIST FORMULA, INCLUDING UNCONDIT FORMULAE */

lf(Fno):-
list_formula(Fno,910101,Value_of_formula).

list_formula(Fno,Date,Value_of_formula):-
formula(Fno,LHS_formula_no,Condition,RHS_formula_no,True_formula_no,
False_formula_no),
(var(LHS_formula_no),
var(Condition),

Formula - 2
```prolog
49  var(RHS_formula_no),
50  var(False_formula_no),
51  list_uncondit_formula(True_formula_no),
52  (var(Date),true
53  ;
54  evaluate_formula(Fno,Date,Value_of_formula),
55  tab(10),write("••• On date "),write(Date),write(" formula evaluates to "),
56  write(Value_of_formula),write(" •••"),nl
57 )
58 ;
59  nonvar(LHS_formula_no),
60  nonvar(Condition),
61  nonvar(RHS_formula_no),
62  nonvar(True_formula_no),
63  nonvar(False_formula_no),
64  tab(10),write( formula(Fno,LHS_formula_no,Condition,RHS_formula_no,True_formula_no,
65       False_formula_no)),nl,
66  list_uncondit_formula(LHS_formula_no),
67  list_uncondit_formula(RHS_formula_no),
68  list_uncondit_formula(True_formula_no),
69  list_uncondit_formula(False_formula_no),
70  (var(Date),true
71  ;
72  evaluate_formula(Fno,Date,Value_of_formula),
73  tab(10),write("••• On date "),write(Date),write(" formula evaluates to "),
74  write(Value_of_formula),write(" •••"),nl
75 )
76 ;
77  write("Error: invalid conditions in formula"),nl
78 ).
79 
80 luf(UFno):-
81  list_uncondit_formula(UFno).
82
83 list_uncondit_formula(UFno):-
84  uncondit_formula(UFno,Coeff,Term_list),
85  tab(10),write(uncondit_formula(UFno,Coeff,Term_list)),nl,
86  fail.
87  list_uncondit_formula(UFno).
88
89 evaluate_formula(Formula_no,Date,Value_of_formula):-
90  formula(Formula_no,LHS_formula_no,Condition,RHS_formula_no,True_formula_no,
91       False_formula_no),
92  /* write("Formula is "),tab(2),write("/
93  /* formula(Formula_no,LHS_formula_no,Condition,RHS_formula_no,True_formula_no,"/
94  /* False_formula_no)),nl, */
95  (var(LHS_formula_no),
96   var(Condition),
97   var(RHS_formula_no),
98   var(False_formula_no),
99   evaluate_uncondit_formula(True_formula_no,Date,Value_of_formula)
100 ;
101  nonvar(LHS_formula_no),
102  nonvar(Condition),
103  nonvar(RHS_formula_no),
104       False_formula_no))
105 ).
106
107 evaluate_uncondit_formula(Formula_no,Date,Value_of_formula):-
108  formula(Formula_no,LHS_formula_no,Condition,RHS_formula_no,True_formula_no,
109       False_formula_no),
110  var(LHS_formula_no),
111  var(RHS_formula_no),
112  var(False_formula_no),
113  evaluate_uncondit_formula(True_formula_no,Date,Value_of_formula)
114 ;
115  nonvar(LHS_formula_no),
116  nonvar(Condition),
117  nonvar(RHS_formula_no),
118       False_formula_no))
119 ).
```

Formula - 3
evaluate_uncondit_formula(LHS_formula_no,Date,Value_of_LHS_formula),
evaluate_uncondit_formula(RHS_formula_no,Date,Value_of_RHS_formula),
/*
write('Condition=',write(Condition),nl,'*/
/*
write(Value_of_LHS_formula),tab(2),write(Value_of_RHS_formula),nl,'*/
once((
  Condition(Value_of_LHS_formula,Value_of_RHS_formula),
evaluate_uncondit_formula(True_formula_no,Date,Value_of_formula),
/*
write('Value of true formula='),write(Value_of_formula),nl,'*/
true
))
/*
write('Value of false formula='),write(Value_of_formula),nl,'*/
true
)
write('Error. invalid conditions in formula'),nl,
write(formula(Formula_no,LHS_formula_no,Condit,RHS_formula_no,True_formula_no,

Value_of_formula is 0).
evaluate_uncondit_formula(Uncondit_formula_no,Date,Value_of_uncondit_formula):-
/*
write('Entering evaluate_uncondit_formula for formula '),n/*
write(Uncondit_formula_no),nl,5/
sum(Value Returned,
  ( uncondit_formula(Uncondit_formula_no,Coeff,Term_list),
    once((evaluate_term_list(Coeff,Date,1,Value Returned,Term_list)))
  ),Value_of_uncondit_formula),
/*
write('Value of uncondit_formula is '),write(Value_of_uncondit_formula),nl,'*/
true.
evaluate_term_list(Coeff,Date,Value_in,Value Returned,[HIT]):-
/*
write('evaluate uncondit_formula, H is'),write(H),nl,*/
[Variable, Power] = H,
/*
write('Power ='),write(Power),nl,'*/
(P_formula_no = Power,
/*
write('P_formula_no = '),write(P_formula_no),nl,'*/
/* exponential term is itself an uncondit. formula */
evaluate_uncondit_formula(P_formula_no,Date,Value_of_exp)
/*
write('Variable ='),write(Variable),nl,'*/
once(( Variable == d,
  reference_date_for_formula(Ref_date),
  number_of_days_from_to(Ref_date,Date,Days),
  /*
  write('number of days=',write(Days),nl,'*/
  Intermediate_value is Value_in * (Days)^Value_of_exp
  )
  [V_formula_no] = Variable,
/* variable term is itself an unconditional formula */
evaluate_uncondit_formula(V_formula_no,Date,Value_of_var),
/*
write('Value of true formula='),write(Value_of_formula),nl,'*/
true
))
/*
write('Value of false formula='),write(Value_of_formula),nl,'*/
true
)
write('Error. invalid conditions in formula'),nl,
write(formula(Formula_no,LHS_formula_no,Condit,RHS_formula_no,True_formula_no,

Value_of_formula is 0).

Formula - 4
Intermediate_value is Value_in^Value_of_var^Value_of_exp

Intermediate_value is Value_in^Variable^Value_of_exp

gl_account(GL_ACno) = Variable, /* have a ref to another a/c */
evaluate_ac_balance(GL_ACno,Date,Balance),
Intermediate_value is Value_in^Value_of_exp

date_function(D,Y,price_index(Variable,D,Y),Date,F_value),
Intermediate_value is Value_in^F_value^Value_of_exp

write('Can’t evaluate variable in evaluate_term_list, term='),
write(H),nl

evaluate_term_list(Coeff,Date,Intermediate_value,Value_returned,T).

reference_date_for_formulae(000101).

evaluate_ac_balance(GL_ACno,Date,Balance):-
/* loops through the balance formulae using evaluate_formula */
retractall(eval_balance_formula_value(GL_ACno,_)),
eval_balance_formulae(GL_ACno,Date,Balance).

eval_balance_formulae(GL_ACno,Date,Balance):-

gl_ac_balance_formula(GL_ACno,Fno),
once((
evaluate_formula(Fno,Date,Value_of_formula),
assertz(eval_balanceformula_value(GL ACno,Value_of_formula))
)),fail.
eval_balance_formulae(GL_ACno,Date,Balance):-
cnt(Value_of_formula,eval_balance_formula_value(GL_ACno,Value_of_formula),Count),
( Count @>= 1,
sum(Value_of_formula,eval_balance_formula_value(GL_ACno,Value_of_formula),Balance)
/* tab(10),write('### Account balance on date '),write(Date), */
/* write(' is '),write(Balance),nl */
); Balance is 0
),
retractall(eval_balance_formula_value(GL_ACno,_));

/* ----- LISTING OF LEDGER ACCOUNT BALANCES, TRIAL BALANCE ---------*/
tb:-
report_date(Report_date),
tb(Report_date).

tb(Report_date):-
once(
( start_accounting_period(Start_accounting_period);true),
nl,write('Trial Balance for REEFA LTD, date '),write(Report_date),
price_level_accounting_switch(S1),
ifo_lifo_switch(S2),
/* leased_assets_on_off_balance_sheet(S3), */
/* write('for '),write(S1),write(''),write(S2),write(' and leased assets '), */
/* write(S3),write(' balance sheet.'),nl nl */
write(' for '),write(S1),write(''),write(S2),nl,nl,
Formula - 5
retractall(ledger_account_balance(_))

chart_of_accounts(GL_ACno,Description,Class),
once(gl_ac_balance_formula(GL_ACno,Fno)), /* to check that balance exists*/
once(list_bal(GL_ACno,Report_date,Balance)),
fail.

tb(Report_date):-
sum(Balance,ledger_account_balance(Balance),Ledger_balance),
write("Overall ledger balance on date ",write(Report_date),
write(ledger_account_balance(Balance)),nl,
write("
"),nl).

list_bal(GL_ACno):-
report_date(Report_date),
list_bal(GL_ACno,Report_date,Balance).

list_bal(GL_ACno,Report_date,Balance):-
chart_of_accounts(GL_ACno,Description,Class),
tab(3),write("GL_AC = "),write(GL_ACno),write("
"),
write("<"),write_fixed_length(Description,45),write(">
"),nl,
retractall(balance_formula_value(GL_ACno,_)),
list_balance_formulae(GL_ACno,Report_date,Balance).

list_balance_formulae(GL_ACno,Report_date,Balance):-
 gl_ac_balance_formula(GL_ACno,Fno),
 once((', list_formula(Fno,Report_date,Value_of_formula),
 assertz(ledger_account_balance(Value_of_formula)),
 assertz(balance_formula_value(GL_ACno,Value_of_formula))
 ),
 fail.

list_balance_formulae(GL_ACno,Report_date,Balance):-
cnt(Value_of_formula,balance_formula_value(GL_ACno,Value_of_formula),Count),
( Count @>= 1,
 sum(Value_of_formula,balance_formula_value(GL_ACno,Value_of_formula),Balance),
tab(10),write("Overall account balance on date ",write(Report_date),
 write("
"),write(Balance),nl
 ;
 Balance is 0
 ),
retractall(balance_formula_value(GL_ACno,_)).

/*-------------------------- MONEY TB ---------------*/

money_tb:-
report_date(Report_date),
money_tb(Report_date).
money_tb(Report_date):-
 (start_accounting_period(Start_accounting_period);true),
nl,nl,
write("MONEY TRIAL BALANCE, REEFA LTD, as at ",
write(Report_date),
write("for "),write(S1),write("
"),write(S2),nl,nl,

Formula - 6
/* write(' and leased assets '), */
write('Account Description DR Balance CR Balance'),nl,
write(' (cents) (cents)'),nl,
write('--------------------------- ------------'),nl,
retractall(debit_total(_)),
retractall(credit_total(_)),
assert(debit_total(0)),
assert(credit_total(0)),
print_the_detail_for_tb(Report_date), /* this is where all the work is done*/
debit_total(Total_debits),
credit_total(Totalredits),
tab(55),write('---------- ----------'),nl,
tab(55),write(Total_debits),tab(5),write(Total_credits),nl,
tab(55),write('============= ============'),nl,
Positive_value_of_total_credits is - Total_credits,
( Total_debits = Positive_value_of_total_credits
;
    Total_debits \= Positive_value_of_total_credits,
write('************************************************************************ERROR **************************************************************************'),
nl,nl,
write('TRIAL BALANCE DOES NOT BALANCE. '),nl,
Diff is Total_debits + Total_credits,
write('Difference is '),write(Diff),write(' cents.'),nl,
write('************************************************************************ERROR **************************************************************************'),
nl,nl,
print_the_detail_for_tb(Date):-
chart_of_accounts(GL_ACno,Description,Class),
once(
    ( once(evaluate_ac_balance(GL_ACno,Date,Balance)),
      once(round_X_to_two_dec_places(Balance,Rounded_balance)),
      Rounded_balance \= 0,
      write(GL_ACno),tab(3),
      write(Rounded_balance),nl,
      retract(debit_total(T)),
      New_T is T + Rounded_balance,
      asserta(debit_total(New_T)),
      write(Rounded_balance),nl
    ;
    Rounded_balance < 0,
    retract(credit_total(T)),
    New_T is T + Rounded_balance,
    asserta(credit_total(New_T)),
    tab(12),write(Rounded_balance),nl
  )
).
print_the_detail_for_tb(Date).

Formula - 7
/* ------------------ GENERATING A MONEY "T-ACCOUNT" ------------------ */

tac(GL_ACno):-
  report_date(Report_date),
  tac(GL_ACno,Report_date).

tac(GL_ACno,Report_date):-
  (start_accounting_period(Start_accounting_period);true),
  chart_of_accounts(GL_ACno,Description,Class),
  write('GL A/C '),write(GL_ACno),write(' '),
  write('for period from '),write(Start_accounting_period),write(' to Report date '),
  write(Report_date),nl, nl,
  retractall(debit_total(_)),
  retractall(credit_total(_)),
  assert(debit_total(0)),
  assert(credit_total(0)),
  print_the_detail_for_tac(GL_ACno,Report_date), /* this is where all the work is done */

  n1l, nl,
  debit_total(Total_debits),
  credit_total(Total_credits),
  Balance is Total_debits + Total_credits, /* credits is -ve */
  ( Balance > 0,
    tab(19),write('Balance'),tab(12),write(Balance),nl,
    Total_credits_plus_balance is Total_credits - Balance,
    tab(20),write('-------------'),tab(24),write('--------------'),nl,
    tab(19),write(Total_debits),tab(23),write(Total_credits_plus_balance),nl,
    tab(20),write(' 	 '),tab(24),write(' 	 '),nl,
    tab(39),write('Balance'),tab(13),write(Balance),nl
  ),
  Balance < 0,
  write('Balance'),tab(13),write(Balance),nl,
  Total_debits_plus_balance is Total_debits - Balance,
  tab(20),write('-------------'),tab(24),write('--------------'),nl,
  tab(19),write(Total_debits_plus_balance),tab(23),write(Total_credits),nl,
  tab(20),write(' 	 '),tab(24),write(' 	 '),nl,
  write('Balance'),tab(13),write(Balance),nl
  ),
  Balance == 0,
  tab(20),write('---------'),tab(24),write('--------------'),nl,
  tab(19),write(Total_debits),tab(23),write(Total_credits),nl,
  tab(20),write(' 	 '),tab(24),write(' 	 '),nl,
  write('Balance'),tab(13),write(Balance),nl
).

print_the_detail_for_tac(GL_ACno,Date):-
  jed(JEno,Lno,GL_ACno,Fno),
  once(evaluate_formula(Fno,Date,Amount)),
  once((

Formula - 8
to get Eno*/
( Amount >= 0,
  retract(debit_total(T)),
  New_T is T + Amount,
  assert(debit_total(New_T)),
  write(JEno),write(" "),write(Eno),tab(14),write(Amount),nl
  ;
Amount < 0,
  retract(credit_total(T)),
  New_T is T + Amount,
  assert(credit_total(New_T)),
  tab(39),write(JEno),write(" "),write(Eno),tab(14),write(Amount),nl
 })
 fail.
print_the_detail_for_tac(GL_ACno,Date).
/* alltac writes out all T-accounts that that have one or more journal entries*/
alltac:-fast_setof(GL_ACno,jed(_,GL_ACno,:),ACs),
  write('ACs='),write(ACs),nl,
dotac(ACs).
dotac([]),
dotac([H | T]):-
  once(tac(H)),
  nl,
dotac(T).

/*-------------- CHECK THAT A GIVEN JE BALANCES ---------------*/
check_balance(JEno):-
  check_balance(JEno,'Write error messages').
check_balance(JEno,E or B):-
  /* E or B is for 'Write error messages' or 'Balance formulae required'. */
  /* The latter uses assertz */
  /* balance_formula_reqd(JEno,Test,Balance_coeff,Term,TrueFalse) */
  once((retractall(balance_formula_reqd(_,_,_,_,_)))),
  /* write("balance_formula_reqd retracted"),nl, */
  fast_setof(Test,( jed(JEno,_,_,Fno),
    once((get_test_in_formula(Fno,Test)))), Tests),
  /* write("About to check balances for formula with tests [[LHS,C,RHS,... ]]

  check_balance_for_each_test(JEno,E or B,Tests).
get_test_in_formula(Fno,Test):-
  formula(Fno,LHS UFno,C,RHS UFno,True UFno,False UFno),
  once(( var(LHS UFno),var(C),var(RHS UFno),
    Test=[])
  ;
    nonvar(LHS UFno),
    nonvar(C),
    nonvar(RHS UFno),
    get_all_terms_in_uncondit_formula(LHS UFno,LTerms),
    get_all_terms_in_uncondit_formula(RHS UFno,RTerms),
    Test=[LTerms,C,RTerms]
  )
  ).

Formula - 9
410; write('Error: invalid test in check je balances'), 
411write(' test ignored.'), nl 
412)).

get_all_terms_in_uncondit_formula(UFno, Terms) :- 
414fast_setof(Term, uncondit_formula(UFno, Coeff, Term), Setof_Terms), 
416sum_coeff_of_each_term(UFno, Setof_Terms, [], Terms). 
417/* write('List of terms for conditional formula '), write(UFno), write( '='), nl, */ 
418/* write(Terms), nl */ 

419sum_coeff_of_each_term(UFno, [ ], Terms). 
420sum_coeff_of_each_term(UFno, [ H | T ], Shorter_list, Terms) :- 
421sum(Coeff, uncondit_formula(UFno, Coeff, H), Sum_of_coeff), 
422/* write('Term='), write(H), write( ' and coeff='), write(Sum_of_coeff), nl, */ 
423append([[Sum_of_coeff, H]], Shorter_list, Longer_list), 
424/* write('Longer list= '), write(Longer_list), nl, */ 
425sum_coeff_of_each_term(UFno, T, Longer_list, Terms).

check_balance_for_each_test(JEno, E or B, []). 
check_balance_for_each_test(JEno, E or B, [H | T ]):- 
428/* write('Test to be checked for balance = '), write(H), nl, */ 
429/* write('---------------------'), nl, */ 
430(H = []), 
431/* check balance of unconditional part of je */ 
432fast_setof(Term, ( jed(JEno, _ , Fno)), 
433formula(Fno, LHS_UFno, C, RHS_UFno, True_UFno, False_UFno), 
434var(LHS_UFno), 
435var(C), 
436var(RHS_UFno), 
437uncondit_formula(True_UFno, Coeff, Term)), Terms), 
438/* write('About to check balances for test [] with terms '), */ 
439/* write('as follows:'), nl, write(Terms), nl, */ 
440check_balance_for_each_true_term(JEno, E or B, H, Terms) 
441; 
442/* If=[]. */ 
443/* must have a conditional formula, need to check true and false UF nos */ 
444/* check balance of conditionally true part of je */ 
445fast_setof(TTerm, ( jed(JEno, _ , Fno)), 
446formula(Fno, LHS_UFno, C, RHS_UFno, True_UFno, False_UFno), 
447get_all_terms_in_uncondit_formula(LHS_UFno, LTerms), 
448get_all_terms_in_uncondit_formula(RHS_UFno, RTerms), 
449H == [LTerms, C, RTerms], 
450uncondit_formula(True_UFno, Coeff, TTerm)), TTerms), 
451/* write('About to check balances for test '), write(H), nl, */ 
452/* write('with true formula terms as follows:'), nl, write(TTerms), nl, */ 
453check_balance_for_each_true_term(JEno, E or B, H, TTerms), 
454/* now check balance of conditionally false part of je */ 
455fast_setof(FTerm, ( jed(JEno, _ , Fno)), 
456formula(Fno, LHS_UFno, C, RHS_UFno, True_UFno, False_UFno), 
457get_all_terms_in_uncondit_formula(LHS_UFno, LTerms), 
458get_all_terms_in_uncondit_formula(RHS_UFno, RTerms), 
459H == [LTerms, C, RTerms], 
460uncondit_formula(False_UFno, Coeff, FTerm)), FTerms),
/* write('About to check balances for test '),write(H),nl, */
/* write('with false formula terms as follows:'),nl,write(FTerms),nl, */
check_balance_for_each_false_term(JEno,E_or_B,H,FTerms)
)
/* now check for remaining tests in conditional parts of je */
check_balance_for_each_true_test(JEno,E_or_B,T).

check_balance_for_each_true_term(JEno,E_or_B,Test,[ ]).

/* write('Checking balance for conditionally true term in je '),write(H),nl, */
/* write(' term='),write(H),nl, */
sum(Coeff,( jed(JEno,_,_,Fno),
  formula(Fno,LHS_UFno,C,RHS_UFno,True_UFno,False_UFno),
  ( var(LHS_UFno),var(C),var(RHS_UFno),
    Test ==[]
  ),
  uncondit_formula(True_UFno,Coeff,H)), Sum of_coeff),
( E_or_B == 'Balance formulae required',
  Balance_coeff is - Sum of_coeff,
  assertz(balance_formula_reqd(JEno,Test,Balance_coeff,H,t))
);
E_or_B == 'Write error messages',
( Sum of_coeff > -0.0000005,
  Sum of_coeff < 0.0000005/* do nothing */
);
write('Error: je '),write(JEno),write(' does not balance for True term '),write(H);nl,
write(' in test='),write(Test),nl,
write(' Sum of coeff = '),write(Sum of coeff),nl
)
write('Error: Invalid B or E for check je balances'),
write(E_or_B),nl
);
check_balance_for_each_true_term(JEno,E_or_B,Test,T).

check_balance_for_each_false_term(JEno,E_or_B,Test,[ ]).

/* write('Checking balance for conditionally false term in je '),write(JEno),nl, */
/* write(' term='),write(H),nl, */
sum(Coeff,( jed(JEno,_,_,Fno),
  formula(Fno,LHS_UFno,C,RHS_UFno,True_UFno,False_UFno),
  get_all_terms_in_uncondit_formula(LHS_UFno,LTerms),
  get_all_terms_in_uncondit_formula(RHS_UFno,RTerms),
  Test ==[LTerms,C,RTerms]
  ),
  uncondit_formula(False_UFno,Coeff,H)), Sum of_coeff),
( E_or_B == 'Balance formulae required',
  Balance_coeff is - Sum of_coeff,
  assertz(balance_formula_reqd(JEno,Test,Balance_coeff,H,t))
);
E_or_B == 'Write error messages',
( Sum of_coeff > -0.0000005,
  Sum of_coeff < 0.0000005/* do nothing */
)
Formula - 11
write("Error: je "),write(JEno),write(" does not balance for False term "),
write(H),nl,write(" in test="),write(Test),nl,
write(" Sum_of_coeff = "),write(Sum_of_coeff),nl
)
;
write("Error: Invalid B or E for check _je balances"),
write(E_or_B),nl,
check_balance_for_each_false_term(JEno,E_or_B,Test,T).

/*---------------------------------- POST JE TO GLAC BALANCE FORMULAE ----------------------------------*/
tje(JEno):-
/* retractall( gl_ac_balance_formula(_,_)), */
postJe(JEno),
listing(gl_ac_balance_formula).

post_Je(JEno):-
jec(JEno,GL_ACno,Fno),
once(
( once(get_test_in_formula(Fno,JEF_Test)),
 ( gl_ac_balance_formula(GL_ACno,BFno),
 once(get_test_in_formula(BFno,BF_Test)),
 /* write("$ Test in je formula = "),write(JEF_Test),nl, */
 /* write("$ Test in bal formula = "),write(BF_Test),nl, */
 BF_Test == JEF_Test,
 /* add to existing formula */
 formula(Fno,_,JTrue_UFno,JFalse_UFno),
 formula(BFno,BLHS,C,BRHS,BTrue_UFno,BFalse_UFno),
 /* write("About to post formula:");nl, */
 /* write(formula(Fno,_,JTrue_UFno,JFalse_UFno)),nl, */
 /* write( ") to balance formula:");nl,tab(2), */
 /* write(formula(BFno,BLHS,C,BRHS,BTrue_UFno,BFalse_UFno)),nl, */
 /* ( var(C), /* have uncondit formula */
 once(update_uncondit_formula(JTrue_UFno,BTrue_UFno,New_bal_list)),
 once((New_bal_list == [0]),
 /* write("**** have conditional True and False "),nl, */
 once(update_uncondit_formula(BTrue_UFno,BFalse_UFno,New_bal_list)),
 once(update_uncondit_formula(BFalse_UFno,BTrue_UFno,New_F_bal_list)),
 once((New_T_bal_list == [0], New_F_bal_list == [0]),
 /* write(retract(uncondit_formula(BLHS,_,_))),nl, */
 retract(uncondit_formula(BLHS,_,_)),
 /* write(retract(uncondit_formula(BRHS,_,_))),nl, */
 retract(uncondit_formula(BRHS,_,_)),
 /* write(retract(uncondit_formula(GL_ACno,BFno))),nl, */
 retract(gl_ac_balance_formula(GL_ACno,BFno)))
 true
))
;
/* jel matches existing conditional formula, so do True and False */
write("**** have conditional True and False formulae to update ****");nl, */

/Formula - 12
retract(uncondit_formula(BRHS,_,_)),
"  write(retract(uncondit_formula(BTrue_UFno,_,_)),nl,*/
retract(uncondit_formula(BTrue_UFno,_,_)),
"  write(retract(uncondit_formula(BFalse_UFno,_,_)),nl,*/
retract(uncondit_formula(BFalse_UFno,_,_)),
"  write(retract(formula(BFno,BLHS,C,BRHS,BTrue_UFno,BFalse_UFno))),nl,*/
retract(formula(BFno,BLHS,C,BRHS,BTrue_UFno,BFalse_UFno)),
"  write(retract(gl_açbalance_formula(GL_ACno,BFno))),nl,*/
retract(gl_ac_balance_formula(GL_ACno,BFno))

true
)

; new test, so need a new BFno */
formula(Fno,JLHS_UFno,C,JRHS_UFno,JTrue_UFno,JFalse_UFno),

/*
#write('####################################################
##
'write('About to assert new condit balance formula for formula:'),nl, */
write(formula(Fno,JLHS_UFno,C,JRHS_UFno,JTrue_UFno,JFalse_UFno)),nl,*/
assert_condit_balance_formula(GL_ACno,_,BLHS_UFno,C,BRHS_UFno,
BTrue_UFno,BFalse_UFno),
( var(C), /* only True uncondit formula required */
copy_uncondit_formula(JTrue_UFno,BTrue_UFno)
; copy_uncondit_formula(JLHS_UFno,BLHS_UFno),
copy_uncondit_formula(JRHS_UFno,BRHS_UFno),
copy_uncondit_formula(JTrue_UFno,BTrue_UFno),
copy_uncondit_formula(JFalse_UFno,BFalse_UFno)
)
)
)

fail.
postje(JEno).

assert_condit_balance_formula(GL_ACno,Fno,LHS_UFno,C,RHS_UFno,
True_UFno,False_UFno):-
get_next_Fno(Fno),
get_next_UFno(True_UFno),
( var(C) /* do nothing, null condition*/
; get_next_UFno(LHS_UFno),
get_next_UFno(RHS_UFno),
get_next_UFno(False_UFno)
),
assertz(gl_ac_balance_formula(GL_ACno,Fno)),
/* write(gl_ac_balance_formula(GL_ACno,Fno)),nl,*/
assertz(formula(Fno,LHS_UFno,C,RHS_UFno,True_UFno,False_UFno)),
/* write(formula(Fno,LHS_UFno,C,RHS_UFno,True_UFno,False_UFno)),nl,*/
ture.
assert_condit_balance_formula(GL_ACno,Fno,LHS_UFno,C,RHS_UFno,
True_UFno,False_UFno):-
write('Error: assert condit_balance_formula failed, GL_ACno='),
write(GL_ACno),nl.
copy_uncondit_formula(Orig_UFno,Copy_to_UFno):-
    uncondit_formula(Orig_UFno,Orig_Coeff,Orig_Terms),
    once(( assertz(uncondit_formula(Copy_to_UFno,Orig_Coeff,Orig_Terms)),
           /* write('Copying uncondit formula:'),nl, */
           /* write(uncondit_formula(Copy_to_UFno,Orig_Coeff,Orig_Terms)),nl, */
           true
    )),
    fail.
    copy_uncondit_formula(Orig_UFno,Copy_to_UFno).

update_uncondit_formula(J_UFno,B_UFno,New_B_Coeff_list):-
    once((retractall(new_bal coeff(_)))),
    uncondit_formula(J_UFno,J_Coeff,J_Term),
    once(( ( retract(uncondit_formula(B_UFno,B_Coeff,J_Term)),
              /* note J in line above to ensure terms match */
              New_B_Coeff is B_Coeff + J_Coeff,
              assertz(uncondit_formula(B_UFno,New_B_Coeff,J_Term)),
              /* write(uncondit_formula(B_UFno,New_B_Coeff,J_Term)),nl, */
              assertz(new_bal coeff(New_B_Coeff)),
              true
              ),
          assertz(uncondit_formula(B_UFno,J_Coeff,J_Term)),
          /* write(uncondit_formula(B_UFno,J_Coeff,J_Term)),nl, */
          assertz(new_bal coeff(J_Coeff))
    )),
    fail.
    update_uncondit_formula(J_UFno,B_UFno,New_B_Coeff_list):-
    fast_setof(New_B_Coeff,new_bal coeff(New_B_Coeff),New_B_Coeff_list),
    /* write('New balance coefficient list is '),write(New_B_Coeff_list),nl, */
    true.

/* -----------TRANSFER ACCOUNT BALANCES TO PROFIT AND LOSS ----------- */
/* DOES NOT "CLOSE" ACCOUNTS, LEAVES BALANCE FORMULAE INTACT. */

/* Note that this procedure is "impure" in that it uses code from the Multiview
 Accounting Interpreter. Otherwise, this file is self-contained. It refers only
to procedures within the file. */

clacs:-
class(900101).
class(Date):-
    close_revenue_acs(Date).

close_revenue_acs(Date):-
    write('Entering close_revenue_acs for period ending '),write(Date),nl,
    assert(JEno,Date,closing_acs,'Closing revenue and expense for period'),
    map_AC_concept_to_GLac('Profit and Loss',PL_ACno),
    close_GL_accounts(JEno,Date,PL_ACno),
pot_j(EJEno).

close_GL_accounts(JEno,Date,PL_ACno):-
    chart_of_accounts(GL_ACno,_,Class),
    once(( (is_a(glAC_hierarchy,Class,'Operating Income');
              is_a(glAC_hierarchy,Class,'Non-operating Income')),
          gl_ac_balance_formula(GL_ACno), /* only assert if bal exists */
          true.
    )}.
assert_closing_jel(JEno, Date, GL_ACno, PL_ACno),
fail.
close_GL_accounts(JEno, Date, PL_ACno).

assert_closing_jel(JEno, Date, GL_ACno, PL_ACno):-

    gl_ac_balance_formula(GL_ACno, Fno), /* may be more than one */
    once((
        evaluate_formula(Fno, Date, F_value),
        indexed_assert_jel_P_and_L_item(JEno, PL_ACno, F_value, Date),
        Minus_F_value is - F_value,
        price_level_accounting_switch(PLA_type),
        PLA_type == 'Current Cost Accounting',
        get_Rtype_for_this_GLac(GL_ACno, Rtype),
        nonvar(Rtype),
        indexed_assert_jel_Rtype(JEno, GL_ACno, Minus_F_value, Date, Rtype)
    ),
    indexed_assert_jel_P_and_L_item(JEno, GL_ACno, Minus_F_value, Date)
    ),
fail.
assert_closing_jel(JEno, Date, GL_ACno, PL_ACno).

get_Rtype_for_this_GLac(GL_ACno, Rtype):-
    map_AOac_to_GLac(AO_ACno, GL_ACno),
    nl, write('never expect to find an expense account'),
    write(' which maps directly to an AOac '),
    write(map_AOac_to_GLac(AO_ACno, GL_ACno)),
    nl, write('This message comes from get_Rtype_for_this_GLac in '),
    write('close_revenue_acs'),
    nl
    ;
    gl_depreciation(Dr_ACno, _, GL_ACno),
    map_AOac_to_GLac(AO_ACno, Dr_ACno)
    ;
    gl_amortization(Dr_ACno, GL_ACno),
    map_AOac_to_GLac(AO_ACno, Dr_ACno)
    ),
    ao_account(AO_ACno, _, Class, Rno, Agent_id, ),
    resource(Rno, _, Rtype),
    /* write(Rtype for GL_ACno '),write(GL_ACno),write(' is '), */
    /* write(Rtype),nl */
true.

get_Rtype_for_this_GLac(GL_ACno, Rtype).

done(close_revenue_acs).

/* --------- CREATE JELS TO CLOSE EXISTING ACCOUNT BALANCE FORMULAE */

/* This routine REALLY CLOSES a GL account. When completed, the balance
formula for the GL account is null. The balance will be in Post_to_GLac */
clac(GL_ACno):-
assert jets to close account(GL ACno,Post to ACno,JEno).

assert jets to close account(GL ACno,Post to ACno,JEno):-
  gl ac balance formula(GL ACno,Fno),
  once(assert jel to close this formula(Fno,GL ACno,Post to ACno,JEno)),
  fail.
assert jets to close account(GL ACno,Post to ACno,JEno).

assert jel to close this formula(Fno,GL ACno,Post to ACno,JEno):-
  formula(Fno,LHS UFno,C,RHS UFno,True UFno,False UFno),
  once(( var(LHS UFno),var(C),var(RHS UFno),
      /* have an unconditional formula */
      assert uncondit jel(JEno,GL ACno,_,True UFno),
      reverse uncondit formula(True UFno,True UFno),
      ( nonvar(Post to ACno),
      assert uncondit jel(JEno,Post to ACno,_,PTrue UFno),
      copy uncondit formula(True UFno,PTrue UFno)
      /* above proc is used in post je */
    )
  )
  ;
  true
  ;
  nonvar(LHS UFno),
  nonvar(C),
  nonvar(RHS UFno),
  assert condit jel(JEno,GL ACno,JFno,JLHS UFno,C,JRHS UFno,
      JTrue UFno,JFalse UFno),
  copy uncondit formula(LHS UFno,JLHS UFno),
  copy uncondit formula(RHS UFno,JRHS UFno),
  reverse uncondit formula(True UFno,JTrue UFno),
  reverse uncondit formula(False UFno,JFalse UFno),
  ( nonvar(Post to ACno),
  assert condit jel(JEno,Post to ACno,_,PLHS UFno,C,PRHS UFno,
      PTrue UFno,PFalse UFno),
  copy uncondit formula(LHS UFno,PLHS UFno),
  copy uncondit formula(RHS UFno,PRHS UFno),
  copy uncondit formula(True UFno,PTrue UFno),
  copy uncondit formula(False UFno,PFalse UFno)
  )
  ;
  true
  ;
  write('Error: invalid assert jel to close this formula:'),nl,
  write(formula(Fno,LHS UFno,C,RHS UFno,True UFno,False UFno)),nl

)).

assert jel to close this formula(Fno,GL ACno,Post to ACno,JEno).

/* based on copy uncondit formula in post je */
reverse uncondit formula(Orig UFno,Reversed UFno):-
  uncondit formula(Orig UFno,Orig Coeff,Orig Terms),
  Reversed Coeff is -Orig Coeff,
  once(( assertz(uncondit formula(Reversed UFno,Reversed Coeff,Orig Terms)),
      /* write(Reversing uncondit formula:'),nl, */
      /* write(uncondit formula(Reversed UFno,Reversed Coeff,Orig Terms)),nl, */
      true
    )).
fail.
reverse_uncondit_formula(Orig_UFno,Reversed_UFno).

/
  ASSERT NEW FORMULA, UNCONDIT FORMULA

assert_uncondit镕(JEno,GL_ACno,Fno,UFno):-
  get_next_Fno(Fno),
  get_next_UFno(UFno),
  get_next_je_line_no(L),
  assertz(jed(JEno,L,GL_ACno,Fno)),
  assertz(formula(Fno,_,_,_,UFno,_)),
  /* write(assertz(jed(JEno,L,GL_ACno,Fno))),nl, */
  /* write(assertz(formula(Fno,_,_,_,UFno,_))),nl, */
  true.
assert_uncondit镕(JEno,GL_ACno,Fno,UFno):-
  write('Error: assert uncondit镕 failed, JEno='),write(JEno),
  write(' GL ACno='),write(GL_ACno),write(' Fno='),write(Fno),
  write(' UFno='),write(UFno),nl.

assert_condit镕(JEno,GL_ACno,Fno,LHS_UFno,C,RHS_UFno,True_UFno,False_UFno):-
  get_next_Fno(Fno),
  get_next_UFno(LHS_UFno),
  get_next_UFno(RHS_UFno),
  get_next_UFno(True_UFno),
  get_next_UFno(False_UFno),
  get_next_je_line_no(L),
  assertz(jed(JEno,L,GL_ACno,Fno)),
  assertz(formula(Fno,LHS_UFno,C,RHS_UFno,True_UFno,False_UFno)).
assert_condit镕(JEno,GL_ACno,Fno,LHS_UFno,C,RHS_UFno,True_UFno,False_UFno):-
  write('Error: assert condit镕 failed, JEno='),write(JEno),tab(2),
  write(GL_ACno),nl.

next_je_line_number(1).

get_next_je_line_no(L):-
  retract(next_je_line_number(L)),
  L1 is L + 1,
  asserta(next_je_line_number(L1)).

/* this proc is used by assert_jels above, lit could go in formula but OK here*/
get_next_Fno(Fno):-
  retract(next_formula_number(Fno)),
  Fno1 is Fno + 1,
  asserta(next_formula_number(Fno1)).

get_next_UFno(UFno):-
  retract(next_uncondit_formula_number(UFno)),
  UFno1 is UFno + 1,
  asserta(next_uncondit_formula_number(UFno1)).

/*------------------- ASSERT BALANCING JE LINES FOR JE-------------------*/
/* used when a je created by MV Interpreter needs balancing je line */
abjl(JEno):-
assert_balancing_jel_lines(JEno,12345).

assert_balancing_jel_lines(JEno,GL_ACno):-
  once((
    check_balances(JEno,'Balance formulae required'),
    fast_setof(Test,balance_formula_regd(JEno,Test,_,_,J,Tests)),
  )/* write('About to assert balancing jel for Tests'),nl,write(Tests),nl, */
  assert_balancing_jel_list(JEno,GL_ACno,Tests).

assert_balancing_jel_list(JEno,GL_ACno,[]).

assert_balancing_jel_list(JEno,GL_ACno,[H|T]):-
  once((H==[],
    assert_uncondit_jel(JEno,GL_ACno,_UFno),
    assert_uncondit_bal_formulae(JEno,H,UFno,t)
  ),/* write(assert_uncondit_jel(JEno,GL_ACno,_UFno),nl,assert_uncondit_bal_formulae(JEno,H,UFno,t)),*/
  assert_balancing_jel_list(JEno,GL_ACno,T).

assert_uncondit_bal_formulae(JEno,H,UFno,TrueFalse):-
  /* write('Asserting uncondit formula for test '),write(H),nl, */
  /* write(' and UFno='),write(UFno),nl, */
  balance_formula_regd(JEno,H,Balance_coeff,Term,TrueFalse),
  once((assertz(uncondit_formula(UFno,Balance_coeff,Term)))),
  fail.

assert_uncondit_bal_formulae(JEno,H,UFno,TrueFalse).

assert_condit_bal_formulae(H,LHS_UFno,RHS_UFno):-
  /* write('Asserting LHS and RHS condit formula for test '),write(H),nl, */
  /* write(' , LHS_UFno='),write(LHS_UFno),nl, */
  /* write(' and RHS_UFno='),write(RHS_UFno),nl, */
  [LHS_UF_list,C,RHS_UF_list] = H,
  /* write('LHS uncondit formula list='), write(LHS_UF_list),nl, */
  /* write('RHS uncondit formula list='), write(RHS_UF_list),nl, */
  assert_uncondit_bal_form_list(LHS_UFno,LHS_UF_list),
  assert_uncondit_bal_form_list(RHS_UFno,RHS_UF_list).

assert_uncondit_bal_form_list(UFno,[]).

assert_uncondit_bal_form_list(UFno,[H|T]):-
  /* write('Asserting uncondit formula for UFno='),write(UFno),nl, */
  /* write('[Balance_coeff,Term]),write(H),nl, */
  [Balance_coeff,Term] = H,
  once((assertz(uncondit_formula(UFno,Balance_coeff,Term))))),
  /* write(assertz(uncondit_formula(UFno,Balance_coeff,Term))), */
  assert_uncondit_bal_form_list(UFno,T).

/* -------------------TRASH -------------------------------*/

test 1:-

Formula - 18
uncondit_formula(Uncondit_formula_no,Coeff,Term_list),
write('formula found '),write(uncondit_formula(Uncondit_formula_no,Coeff,Term_list)),nl,
once(( evaluate_term_list(Coeff,910323,1,Value_returned,Term_list),
write('Formula evaluates to '),tab(2),write(Value_returned),nl
)),
fail.

test2:-
evaluate_uncondit_formula(1,900326,Value_of_formula),
write('Formula evaluates to '),tab(2),write(Value_of_formula),nl.

test3:-
test3_formula(Form),
once((evaluate_formula(Form,900101,Value_of_formula),
write('Formula evaluates to '),tab(2),write(Value_of_formula),nl
)),
fail.

test4(Date):-
evaluate_formula(9,Date,Value_of_formula),
write('Formula evaluates to '),tab(2),write(Value_of_formula),nl.

/*
uncondit_formula(1,100,[[2],[2]]).
uncondit_formula(2,-32953,[]).
uncondit_formula(2,1,[[d,1]]).
uncondit_formula(1,100,[[2],[2]]).*/

jed(2,-9).
jed(2,-4).
jed(2,-8).
jed(2,-4).
jed(2,-4).

formual(2,4,=,4,1,2).
formula(3,4,=,5,3,2).
formula(4,4,=,2,2).
formula(5,4,>,5,3,2).
formula(6,4,>,5,3,2).
formula(7,4,>,5,3,2).
formula(8,4,<,3,3,2).
formula(9,10,=,<,11,12,13).
uncondit_formula(10,-32953,[]).
uncondit_formula(10,1,[[d,1]]).
uncondit_formula(11,365.25*10,[]).
uncondit_formula(12,36500/(365.25*10)*32953,[]).
uncondit_formula(12,36500/(365.25*10),[[d,1]]).
uncondit_formula(13,36500,[]).

uncondit_formula(1,100^2,[]).
uncondit_formula(2,200,[[s6,2],[s7,3],[s8,1]]).
uncondit_formula(2,300,[[s9,1],[g,2]]).
uncondit_formula(2,100,[]).
uncondit_formula(3,200,[[s6,2],[s7,3],[s8,1]]).
uncondit_formula(3,300,[[s9,1],[g,2]]).
uncondit_formula(4,1,[]).
uncondit_formula(5,2,[]).
*/
done(formula).
Index to the Procedures in Formula

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>a(JEno)</td>
<td>17</td>
</tr>
<tr>
<td>a.iiac</td>
<td>9</td>
</tr>
<tr>
<td>assert_balancing_j_lines(JEno,GL_ACno)</td>
<td>18</td>
</tr>
<tr>
<td>assert_balancing_j_list(JEno, GL_ACno, [H</td>
<td>T])</td>
</tr>
<tr>
<td>assert_closing_jel(JEno, Date, GL_ACno, PL_ACno)</td>
<td>15</td>
</tr>
<tr>
<td>assert_condit_bal_formulae(H, LHS_UFno, RHS_UFno)</td>
<td>13</td>
</tr>
<tr>
<td>assert_condit_balance_formulae(GL_ACno, Fno, LHS_UFno, C, RHS_UFno, True_UF)</td>
<td>17</td>
</tr>
<tr>
<td>assert_jel_to_close_this_formula(Fno, GL_ACno, Post_to_ACno, JEno)</td>
<td>16</td>
</tr>
<tr>
<td>assert_jel_to_close_account(GL_ACno, Post_to_ACno, JEno)</td>
<td>16</td>
</tr>
<tr>
<td>assert_uncondit_bal_form_list(UFno, [H</td>
<td>T])</td>
</tr>
<tr>
<td>assert_uncondit_bal_formulae(JEno, H, UFno, TrueFalse)</td>
<td>18</td>
</tr>
<tr>
<td>assert_uncondit_jel(JEno, GL_ACno, Fno, UFno)</td>
<td>17</td>
</tr>
<tr>
<td>check_balance(JEno)</td>
<td>9</td>
</tr>
<tr>
<td>check_balance_for_each_false_term(JEno, E or B, Test, [H</td>
<td>T])</td>
</tr>
<tr>
<td>check_balance_for_each_test(JEno, E or B, [H</td>
<td>T])</td>
</tr>
<tr>
<td>check_balance_for_each_true_term(JEno, E or B, Test, [H</td>
<td>T])</td>
</tr>
<tr>
<td>check_jel_balances(JEno, E or B)</td>
<td>9</td>
</tr>
<tr>
<td>clac(GL_ACno)</td>
<td>15</td>
</tr>
<tr>
<td>clacs</td>
<td>14</td>
</tr>
<tr>
<td>clacs(Date)</td>
<td>14</td>
</tr>
<tr>
<td>close_GL_accounts(JEno, Date, PL_ACno)</td>
<td>14</td>
</tr>
<tr>
<td>close_revenue_acs(Date)</td>
<td>14</td>
</tr>
<tr>
<td>copy_uncondit_formula(Orig_UFno, Copy_to_UFno)</td>
<td>14</td>
</tr>
<tr>
<td>dotac([H</td>
<td>T])</td>
</tr>
<tr>
<td>eval_balance_formulae(GL_ACno, Date, Balance)</td>
<td>5</td>
</tr>
<tr>
<td>evaluate_ac_balance(GL_ACno, Date, Balance)</td>
<td>5</td>
</tr>
<tr>
<td>evaluate_formula(Formula_no, Date, Value_of_formula)</td>
<td>3</td>
</tr>
<tr>
<td>evaluate_term_list(Coeff, Date, Value_in, Value_returned, [H</td>
<td>T])</td>
</tr>
<tr>
<td>evaluate_term_list(Coeff, Date, Value_in, Value_returned, [])</td>
<td>4</td>
</tr>
<tr>
<td>evaluate_uncondit_formula(Uncondit_formula_no, Date, Value_of_un)</td>
<td>4</td>
</tr>
<tr>
<td>get_all_terms_in_uncondit_formula(UFno, Terms)</td>
<td>10</td>
</tr>
<tr>
<td>get_next_Fno(Fno)</td>
<td>17</td>
</tr>
<tr>
<td>get_next_j_line_no(L)</td>
<td>17</td>
</tr>
<tr>
<td>get_next_UFno(UFno)</td>
<td>17</td>
</tr>
<tr>
<td>get_Rtype_for_this_GLAc(GL_ACno, Rtype)</td>
<td>15</td>
</tr>
<tr>
<td>get_test_in_formula(Fno, Test)</td>
<td>9</td>
</tr>
<tr>
<td>if(Fno)</td>
<td>2</td>
</tr>
<tr>
<td>list_bal(GL_ACno, Report_date, Balance)</td>
<td>6</td>
</tr>
<tr>
<td>list_bal(GL_ACno)</td>
<td>6</td>
</tr>
<tr>
<td>list_balance_formulae(GL_ACno, Report_date, Balance)</td>
<td>6</td>
</tr>
<tr>
<td>list_formula(Fno, Date, Value_of_formula)</td>
<td>2</td>
</tr>
<tr>
<td>list_je(JEno, Report_date)</td>
<td>2</td>
</tr>
<tr>
<td>list_jeds(JEno, Report_date)</td>
<td>2</td>
</tr>
<tr>
<td>list_jes</td>
<td>2</td>
</tr>
<tr>
<td>list_jes(Report_date)</td>
<td>2</td>
</tr>
<tr>
<td>list_uncondit_formula(UFno)</td>
<td>3</td>
</tr>
<tr>
<td>luf(UFno)</td>
<td>3</td>
</tr>
<tr>
<td>money_tb</td>
<td>6</td>
</tr>
<tr>
<td>money_tb(Report_date)</td>
<td>6</td>
</tr>
<tr>
<td>post_ie(JEno)</td>
<td>12</td>
</tr>
<tr>
<td>print_the_detail_for_tac(GL_ACno, Date)</td>
<td>8</td>
</tr>
<tr>
<td>print_the_detail_for_tb(Report_date)</td>
<td>7</td>
</tr>
</tbody>
</table>

Formula - 21
reverse_uncondit_formula(Orig_UFno,Reversed_UFno) .................................. 16
sum_coeff_of_each_term(UFno,[H|T],Shorter_list,Terms) ............................. 10
tac(GL_ACno,Report_date) ......................................................... 8
tac(GL_ACno) ................................................................. 8
tb ................................................................. 5
tx(Report_date) ............................................................. 5, 6
test1 ................................................................. 18
test2 ................................................................. 19
test3 ................................................................. 19
test4(Date) ............................................................. 19
tpje(JEno) ................................................................. 12
update_uncondit_formula(J_UFno,B_UFno,New_B_Coeff_list) .................. 14

Formula - 22
APPENDIX 2: INTERPRETER

This Appendix contains 1391 lines (28 pages) of Prolog code for the core of the prototype Interpreter. There is an index to all procedures on pages 30-31. Most of the first six pages, devoted to initializing the Interpreter, is of little interest. However, the main loop for the Interpreter is lines 58-72, where line 59 retrieves the next event header, line 64 invokes the event handler, line 67 checks the journal entry balances, and line 68 posts the journal entry.

Causal double-entry reasoning, lines 293-407, is performed once, for all events, before processing of events begins. The system makes two passes through the resource-change, rc(..), predicates, and eventually asserts ed(..) predicates with money values on lines 390 or 393. (Hence the procedure name: generate_eds.)

The default event handler, called "simple", is on lines 411 through 425. On line 413 it calls assert_jeh, commencing line 415, to assert the journal entry headers. On line 414 it then calls loop_through_event_detail_lines, commencing line 422. Procedure loop_through... retrieves ed(..) records one-by-one on line 423, and for each, calls assert_core_je_line to generate the journal entry lines. Procedure assert_core_je_line, commencing line 426, determines the money value for resources not valued by causal double-entry (e.g., for inventory sold or for standard costing) on line 431, and initiates assertion of the journal entry line using map_AOac_to_GLac on line 438 (the actual mapping takes place on line 492). If the resource is depreciable or amortizable, line 443 triggers generation of corresponding journal entries. If the ed(..) record is for an inventory acquisition, details of the transaction are saved on line 463.

The procedures discussed above operate at a similar level of abstraction to Interpreter 1 in Chapter 10. Generation of Formula Accounting journal entries takes place in more detailed modules, at the next level down. The next paragraph focuses on one such procedure, indexed_assert_je_l_Rtype, found on lines 508-576. Once asserted in modules such as indexed_assert_je_l_Rtype, the formulae are evaluated by listing routines in the Formula program as illustrated in Chapter 8, Exhibit 8.1.

Historical-cost "asserts" in indexed_assert_je_l_Rtype are made on line 513 and 514. assert_uncondit_je_l, called on line 513 of Interpreter, is part of the Formula Accounting system, and begins on line 774 of the Formula listing. assertz, on line 514 of Interpreter, is a Prolog primitive instruction. Note the empty variable list ([]) in the uncondit_formula on line 514 for Historical Costing. Generation of CPP and CC/CPP journal entries is handled in lines 516-536, and 538-570, respectively. For CC/CPP accounting, line 539 of Interpreter determines the price index series for this resource type, lines 540 and 541 obtain index values for the general and specific price index series at event date. For monetary resources, the fixed money-value journal entry line is asserted on line 546 (empty variable list, []), followed by a "Gearing Gain" journal entry with variable "g" (the CPI) on line 551 and an empty variable list on line 553. All resources that do not pass through the filter on line 542 are assumed to be non-monetary, so values must be indexed by their specific price index series, Series. This is done in line 559. Real holding gains or losses are recognized on lines 567 (variable "g") and 569 (variable "Series").
/*-------------------------------------------*/

MULTIVIEW ACCOUNTING SYSTEM

P. Seddon, University of Melbourne, January, 1991

/*-------------------------------------------*/

mv(Start_accounting_period,Report_date):-
( Start_accounting_period @=< Report_date,
  mv(Start_accounting_period,Report_date)
).

mv:-
  retract(start_accounting_period(Start_accounting_period)),
  retract(report_date(Report_date)),
  mv(Start_accounting_period,Report_date)
  ;
  write('Error. no A/C period or Report date defined'),nl
).

mv(Start_accounting_period,Report_date):-
  once(initialize_model(Start_accounting_period,Report_date)),
  /* now loop through all events from start up to the report date */
  write('Here are the values of the Multiview Interpreter’s switches for this run'),nl,
  write('for accounting period from '),write(Start_accounting_period),write(' to Report date '),write(Report_date),write(':'),nl,
  write('environment("date",D),
  environment("time",T),
  listing(price_level_accounting_switch),
  listing(fifo_lifo_switch),
  listing(leased_assets_on_off_balance_sheet),
 /* write('Closing dates for this run are as follows:'),nl, */
  listing(closing_dates),
  /* write('Listing closing dates for this run'),nl, */
  once(process_events_for_period(Start_period_date,Next_closing_date)),
  once((
    Next_closing_date @=< Report_date,
    close_revenue_acs(Next_closing_date)
  ;
  true
  )),
  fail.

mv(Start_accounting_period,Report_date):-
  write('ALL EVENTS UP TO REPORT DATE NOW PROCESSED'),nl,
  environment("date",D),
  environment("time",T),
write('Run date and time completed: '),write(D),tab(2),write(T),nl,

once( (tb;true), /* list the trial balance for the run */
      (money_tb;true), /* money trial balance for the run, in CENTS */
      list_jes /* list the journal entries from the run */
    )).

/* --- PROCESS_EVENTS_FOR_PERIOD, INCLUDING THE "CASE" STATEMENT ---*/

process_events_for_period(Start_period_date,Next_closing_date):-
  eh(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative),
  once( ( Date @>= Start_period_date,
      Date @< Next_closing_date,
      / e.g., mv(900101,910101)--> processes all events in 1990 */
      once( (process_event(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno);true)),
      once( ( jeh(JEno,_,Eno,_) ) /* sometimes process_event fails, so JEno not avail */
      checkJe_balances(JEno,'Write error messages'),
      postJe(JEno),
      true )
    )
  ),
  fail.
process_events_for_period(Start_period_date,Next_closing_date).

process_event(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno):-
  map_etype_to_eproc(Etype,Eproc),
  /* avoids backtrack if Eproc fails */
  ( Eproc == no_journal_entry_required,
    write('No je required for event '),write(Eno),
    write(' of type '),write(Etype),nl,
    Eproc(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno) )
  ;
  Eproc \== no_journal_entry_required,
  write('CALLING '),write(Eproc),write(' for event '),write(Eno),
  write(' of type '),write(Etype),nl,
  Eproc(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno)
}).

/* ----- INITIALIZE MODEL ------------------------------*/

initialize_model(Start_accounting_period,Report_date):-
  write('Beginning initialization'),nl,
  ram, /* reset accounting model */
  set_up_dates(Start_accounting_period,Report_date),
  listing(closing_dates),
  riv, /* remove old inventory valuation facts */
  ( retractall(saved_function(_,_,_,_,_));true ),
  ( retractall(payroll_history(_,_,_,_,_,_,_));true ), /*
  ( retractall(recent_payroll_history(_,_,_,_,_,_,_));true ), */
  ( retractall(accrued_wages_dates(_,_,_,_,_,_,_));true ),
  ( ed(Eno,_,_,_,_ ) /* don't need to regenerate eds */
  Interpeter - 3
write("Here are the present values of the Multiview Interpreter's switches

to Report date "),write(Report_date),write(.'),nl,**/ 
write("------------------")),nl,**/ 
listing(price_level_accounting_switch), 
listing(fifo_lifo_switch), 
listing(variance_analysis_frequency), */ 
listing(leased_assets_on_off_balance_sheet), */ 
write("Change switches? Enter y. or n. (in lower case followed by a dot):"),nl, 
once((read(YN))), 
( YN == y, 
once(set_price_level_accounting_switch), 
once(set_fifo_lifo_switch) 
/once(set_variance_analysis_frequency), */ 
/once(set_leased_assets_on_off_balance_sheet_switch) */ 
); 

YN \= y, 
true 
( derived_unit_standard_cost(Rno,____)/* use existing calculations*/ 
( fifo_lifo_switch(standard_cost), 
calc deriving standard costs 
); 
true /* no need to derive std costs */ 
) 
( fifo_lifo_switch(standard_cost), 
/* test formula cost allocation from a/c2802 to 2810, 2820, 2830*/ 
assert Jeh(JEno,Start accounting_period,-1,'Test cost allocation'), 
assert uncondit jeh(JEno,2810,Fno1,UFno1),/* asserts the FORMULA*/ 
assertz(uncondit formula(UFno1,0.6,[gl account(2802),1]]), 
assert uncondit jeh(JEno,2820,Fno2,UFno2), 
assertz(uncondit formula(UFno2,0.3,[gl account(2802),1]]), 
assert uncondit jeh(JEno,2830,Fno3,UFno3), 
assertz(uncondit formula(UFno3,0.1,[gl account(2802),1]]), 
assert uncondit jeh(JEno,2803,Fno4,UFno4), 
assertz(uncondit formula(UFno4,-1,[gl account(2802),1]]), 
post je(JEno) 
); 
true /*no cost allocation je required*/ 
); 
/------------------ RETRACTS and INITIALIZATION --------------------*/ 
ram:- 
retractall(jeh(____)), 
retractall(jed(____)), 
retractall(gl ac_balance_formula(____)), 
retractall(formula(____)), 
retractall(uncondit formula(____)), 

Interpreter - 4
retractall(saved_function(_,_)),
retractall(next_je_number(_)),
retractall(next_je_line_number(_)),
retractall(next_formula_number(_)),
retractall(next_uncondit_formula_number(_)),
asserta(next_je_number(0)),
asserta(next_je_line_number(0)),
asserta(next_formula_number(100)),
asserta(next_uncondit_formula_number(100)),
/* set default switches for Interpreter */
(price_level_accounting_switch(PLA) /*use existing definition*/
  asserta(price_level_accounting_switch('Historical Cost')))
(fifo_lifo_switch(FLS) /* use existing definition*/
  asserta(fifo_lifo_switch(standard_cost))
listing(jeh),
listing(jed). /* listings, jeh & jed, should print two blank lines */
/* NEED TO SORT THE eh table??????? */
/* REMAINDER OF PROCEDURE COMMENTED OUT, 6 DEC 90 */
find_and_assert_accrued_wages_dates(Start_accounting_period,Report_date),
told.
/*
--- DIAGNOSTIC TOOLS --------------------------------------*/
riv:-
listing(inventory_value),
retractall(inventory_value(_)),
listing(inventory_value).
liv:-listing(inventory_value).
red:-
retractall(derived_unit_standard_cost(_)),
retractall(ed(_)).
/* ------- SET_UP_DATES FOR CLOSING REVENUE ACCOUNTS --------*/
set_up_dates(Start_accounting_period,Report_date):-
/* sets up closing dates each year from Start_accounting_period */
retractall(report_date(_)),
retractall(start_accounting_period(_)),
retractall(closing_dates(_)),
asserta(report_date(Report_date)),
asserta(start_accounting_period(Start_accounting_period)),
set_up_closing_dates(Start_accounting_period,Report_date).
set_up_closing_dates(Start_of_period,Report_date):-
Next_closing_date is Start_of_period + 10000,
( Next_closing_date @< Report_date,
assertz(closing_dates(Start_of_period,Next_closing_date)),
/**
  write(closing_dates(Start_of_period,Next_closing_date)),nl, */
set_up_closing_dates(Next_closing_date,Report_date)
; assertz(closing_dates(Start_of_period,Report_date))
).

/*—— SET PRICE LEVEL ACCOUNTING SWITCH ————————*/

set_price_level_accounting_switch:-
write("This Interpreter handles Historical Cost (hc), CPP (cpp) or CC/CPP (cc/cpp) accounting."),nl,
write("Enter choice, either hc, cpp. or cc/cpp. (in lower case followed by a dot):"),nl,
read(PLA_type),
( PLA_type == hc,
  retractall(price_level_accounting_switch(_)),
  asserta(price_level_accounting_switch('Historical Cost'))
;
PLA_type == cpp,
  retractall(price_level_accounting_switch(_)),
  asserta(price_level_accounting_switch('Current Purchasing Power'))
;
PLA_type == cc/cpp,
  retractall(price_level_accounting_switch(_)),
  asserta(price_level_accounting_switch('Current Cost Accounting'))
;
PLA_type \= hc,PLA_type \= cpp,PLA_type \= cc/cpp,
write('Error: you have entered: '),
write(PLA_type),
write('. You must enter either "hc," "cpp." or "cc/cpp.". Retry.'),nl,nl,
set_price_level_accounting_switch
).

/*—— SET FIFO - LIFO SWITCH ————————*/

set_fifo_lifo_switch:-
write("This Interpreter handles either fifo (f), lifo (l) or std. cost (s) inventory valuation."),nl,
write("Enter choice, either f., l. or s. (in lower case followed by a dot):"),nl,
read(FL),
( FL == f,
  retractall(fifo_lifo_switch(_)),
  asserta(fifo_lifo_switch(fifo))
;
FL == l,
  retractall(fifo_lifo_switch(_)),
  asserta(fifo_lifo_switch(lifo))
;
FL == s,
  retractall(fifo_lifo_switch(_)),
  asserta(fifo_lifo_switch(standard_cost))
;
FL\=f,FL\=l,FL\=s,
write('Error: you have entered: '),
248 write(FL),
249 write('. You must enter either "f", "l." or "s.". Retry.\n'), nl, nl,
250 set_fifo_lifo_switch
251 ).

252 /----- SET VARIANCE ANALYSIS FREQUENCY ----------------------*/

253 set_variance_analysis_frequency:-
254 write("This Interpreter handles either annual (a) or monthly (m) variance_analysis.\n"), nl,
255 write("Enter choice, either a. or m. (in lower case followed by a dot):\n"), nl,
256 read(AM),
257 ( AM == a,
258 retractall(variance_analysis_frequency(_)),
259 asserta(variance_analysis_frequency(annual))
260 ;
261 AM == m,
262 retractall(variance_analysis_frequency(_)),
263 asserta(variance_analysis_frequency(monthly))
264 ;
265 AM\=a,AM\=m,
266 write("Error: you have entered: \n"),
267 write(AM),
268 write('. You must enter either "a." or "m.". Retry.\n"), nl, nl,
269 set_variance_analysis_frequency
270 ).

271 /----- SET LEASING OFF BALANCE SHEET SWITCH -------------------*/

272 set_leased_assets_on_off_balance_sheet_switch:-
273 write("This Interpreter handles leases as either on (on) or off (off) balance sheet assets.\n"), nl,
274 write("Enter choice, either on. or off. (in lower case followed by a dot):\n"), nl,
275 read(ON_OFF),
276 ( ON_OFF == on,
277 retractall(leased_assets_on_off_balance_sheet(_)),
278 asserta(leased_assets_on_off_balance_sheet(on))
279 ;
280 ON_OFF == off,
281 retractall(leased_assets_on_off_balance_sheet(_)),
282 asserta(leased_assets_on_off_balance_sheet(off))
283 ;
284 ON_OFF\=on,ON_OFF\=off,
285 write("Error: you have entered: \n"),
286 write(ON_OFF),
287 write('. You must enter either "on." or "off.". Retry.\n"), nl, nl,
288 set_leased_assets_on_off_balance_sheet_switch
289 ).

290 /*-----------------------------------------------------------------*/
291 /* ------------------- GENERATE EDS (CAUSAL DOUBLE-ENTRY)---------*/
292 /*-----------------------------------------------------------------*/

293 generate_eds:-
294 once(retractall(edv(_,_)),
295 once(retractall(ed(_,_,_)));

Interpreter - 7
write('Now using money values exchanged to value non-monetary assets and obligations'),nl,
write('except for the following values accepted from input:

\[
\text{eh(Eno,Date,Etype,_,_,_,_,
\text{eh(Eno,Date,Etype,_,_,_,_,
\text{eh(Eno,Date,Etype,_,_,_,_,
\text{retractall(resource_in_rc(_)),
\text{retractall(resource_in_edv(_))
\text{rc(Eno,Lno,AO_ACno,Q,V),
\text{tab(5),write(rc(Eno,Lno,AO_ACno,Q,V)),nl, */
\text{ao account(AO_ACno_,Class,Rno,_,_),
\text{resource(Rno,_,Unit of measure,Rtype),
\text{assertz(resource_in_rc(Rno)),
\text{member(Rtype,['Cash','Cash at Bank, Current A/C,
\text{Val is Q, /*evaluates ,e.g., 10*5 in Q */
\text{assertz(edv(Eno,Lno,Val))
\text{member(Rtype,['Foreign Currency','Foreign Currency Trade Debt']),
\text{is_a(ao hierarchy,Class,'Asset'),
\text{is_a(ao hierarchy,Class,'Obligation'),
\text{need to BUY $A to convert asset to $A */
\text{need to SELL $A to discharge obligation */
\text{date function(D,Y,foreign_exchange_rate(Unit of measure,D,Y,_),Date,Exchange_rate)
\text{is_a(ao hierarchy,Class,'Asset''),
\text{is_a(ao hierarchy,Class,'Obligation''),
\text{need to BUY $A to convert asset to $A */
\text{need to SELL $A to discharge obligation */
\text{date function(D,Y,foreign_exchange_rate(Unit of measure,D,Y),Date,Exchange_rate)
\text{Val is Q / Exchange_rate,
\text{assertz(edv(Eno,Lno,Val))
\text{member(Etype,['Cash Sale','Credit Sale','Sale of Amortizable Asset'])
\text{no edv asserted, but will assert resource_in_edv below */
\text{Interpreter will calculate its own values, V's mean sales price, not cost */
\text{nonvar(V),
\text{write('Eno '),write(Eno),write(' '),write(Etype),
\text{write(' Lno '),write(Lno),write(' V='),write(V),nl,
\text{assertz(edv(Eno,Lno,V))
\text{resource_in_edv(Rno)),
\text{fail.

Interpreter - 8
value_res(Eno,Date,Etype):-
  /* can only proceed if ONE resource type remains to be valued */
  fast_setof(Red,resource_in_rc(Red),Rs_in_edcs),
  /* write("Rs in edcs = "),write(Rs_in_edcs),nl */
  fast_setof(Redv,resource_in_edv(Redv),Rs_in_edvs),
  /* write("Rs in edvs = "),write(Rs_in_edvs),nl */
  set_diff(Rs_in_edcs,Rs_in_edvs,Unvalued_Rnos),
  /* write("Unvalued_Rnos = "),write(Unvalued_Rnos),nl */
  cardinality(Unvalued_Rnos,C),
  ( C == 0 /* do nothing */
  C==1,
  Unvalued_Rnos = [Rno],
  write('Calculating V for resource: '),write(Rno),nl,
  sum(V,edv(Eno,V),Tot_V),
  sum(Q, ( rc(Eno,AO_ACno,Q),
       ao_account(AO_ACno,Rno),),Tot_Q),
  Unit_val is Tot_V / Tot_Q,
  write('Unit_val is '),write(Unit_val),nl, */
  once((
    rc(Eno,Lno,AO_ACno,Qi),
    once((
      ( ao_account(AO_ACno,Rno),/* only unvalued get through */
      Vi is -Qi * (Tot_V / Tot_Q),
      assertz(edv(Eno,Lno,V))
    )
  )
  fail
  ));
  C > 1,
  write('Error: unable to value more than one Rno in event '),
  write(Eno),nl,
  write('Rnos unvalued are: '),write(Unvalued_Rnos),nl
).

value_res(Eno,Date,Etype).
generate eds(Eno):-
  ed(Eno,Lno,AO_ACno,Q,V),
  /* write(ed(Eno,Lno,AO_ACno,Q,V)),nl */
  once( ( 
    ed(Eno,Lno,Vc),
    assertz(ed(Eno,Lno,AO_ACno,Q,Vc))
    /* write(' '),write(ed(Eno,Lno,AO_ACno,Q,Vc)),nl,nl */
  )
  assertz(ed(Eno,Lno,AO_ACno,Q,V))
  /* write(' '),write(ed(Eno,Lno,AO_ACno,Q,V)),nl,nl */
  assertz(ed(Eno,Lno,AO_ACno,Q,V))
  /* write(' '),write(ed(Eno,Lno,AO_ACno,Q,V)),nl,nl */
  )
},
fail.
generate eds(Eno).
/* (used to compare to pbs entered ed()'s, called ed_old ) */
compare eds(Eno):-
ed(Eno,Lno,AO_ACno,Q,V),
once( ( 
  ed_old(Eno,Lno,AO_ACno,Qo,Vo),
  V \= Vo,
)
write("******"), write(ed(Eno,Lno,AO_Anno,Q,V)), write(" Old V"), write(Vo), nl, nl

compare_eds(Eno).

/* ---------------------------------------------*/

/* CORE INTERPRETER PROCESSING Routines FOLLOW */

simple(Eno,Date,Etype,DocType,Docno,AgentID,Narrative,JEno):-
  write("simple called for event "), write(Eno), nl,
  assert_je(JEno,Date,Eno,Narrative),
  loop_through_event_detail_lines(JEno,Eno,Date). /* always true */

assert_je(JEno,Date,Eno,Narrative):-
  retract(next_je_number(JEno)), /* get next je number */
  Next_JEno is JEno + 1,
  asserta(next_je_number(Next_JEno)), /* record je header*/
  retractall(next_je_line_number(_)), /* set je line number to 1 */
  asserta(next_je_line_number(1)).

loop_through_event_detail_lines(JEno,Eno,Date):-
  ed(Eno,Lno,AO_ACno,Q,V), /* step through the event detail lines */
  assert_core_je_line(Eno,Lno,JEno,Date,AO_ACno,Q,V). /* ends with !, fail. */

loop_through_event_detail_lines(JEno,Eno,Date). /* always true */

assert_core_je_line(Eno,Lno,JEno,Date,AO_ACno,Q,V):-
  write(assert_core_je_line(Eno,Lno,JEno,Date,AO_ACno,Q,V)), nl, /*
  writes je line(s) for one event detail line */
  once(filter_out_conditional_assets_and_obligations(AO_ACno)), */
  once((ao_account(AO_ACno,_,Class, Rno, Agent_id,_, _)),
  once((predictable_future_value(Rno,_,___,xxx),*/
  once((is_a(resource_ev_hierarchy,Rtype,"Shrinking Value Resource"),
  ( Q @> 0,
  generate_acq_depr_amort_jels(JEno,Date,AO_ACno,Q,Money_value,___)
  ;
  true
  /* generate disp depr_amort_jels(JEno,Date,AO_ACno,Q,Money_value,___) */
  /* do nothing, handled in sale_as (which calls this proc */
  /* for all except Q<0 shrinking value resources) */
  )
  ;
  true
  )
})

Interpreter - 10
/* increment inventory_value records if necessary */
/* since the code below may fail, it goes last before fail below */
once((
  Q>0,
  is_a(resource_hierarchy,Rtype,'Unidentified Tangible Resource'),
  not(invalid_Money_value(Money_value)), /* goods ones get through */
  Unit_cost is Money_value/Q,
  once(fifo_lifo_switch(FLS)),
  once( ( 
    ( FLS==fifo; FLS==lifo),
    assertz(inventory_value(Rno,Eno,Date,Unit_cost,Q)),
    write(inventory_value(Rno,Eno,Date,Unit_cost,Q)),nl
  ) ; )
  FLS==standard_cost /* do nothing */
  /* get unit standard_cost_of_resource(Rno,Date,Std_cost,Date_set), */
  /* write(inventory_at_std(Rno,Eno,Date_set,Std_cost,Q)),nl */
))
fail.

/* below is to eliminate Ijiri's "conditional" assets and obligations */
/* not used at present */
filter_out_conditional_assets_and_obligations(AO_ACno):-
  ao_account(AO_ACno,_,Class,Rno,_,_), /* get Class */
  not(is_a(ao_hierarchy,Class,'Conditional Asset')),
  not(is_a(ao_hierarchy,Class,'Conditional Obligation')).
/* only unconditional resource changes get through */
invalid_Money_value(Money_value):-
  /* probably unnecessary now, leave untouched for present */
  Money_value = null,
  !,
  write('Can"t yet handle null Val for event detail'),nl.
invalid_Money_value(Money_value):-
  Money_value<0,
  !,
  write('Negative Value for event detail"'),
  write(Money_value),nl.

/* ASSERT_JEL_VIA_AO_MAPPING -------------------------------*/
assert_jel_via_AO_mapping(JEno,AO_ACno,Money_value,Money_date):-
  /* used when mapping exists and there is no need to check inventory-in */
  map_AOac_to_GLac(AO_ACno,GL_ACno),
  ao_account(AO_ACno,_,Class,Rno,Agent_id,_,),
  resource(Rno,_,Unit_of_measure,Rtype),
  ( member(Rtype,['Foreign Currency','Foreign Currency Trade Debt']),

  foreign_exchange_assert_jel(JEno,GL_ACno,Money_value,Money_date,Class,Unit_of_measure,Q)
  ;
  indexed_assert_jel_Rtype(JEno,GL_ACno,Money_value,Money_date,Rtype)
).
assert_jel_via_AO_mapping(JEno, AO_ACno, Money_value, Money_date); /* get here if above
fails */

not(production_step_input_AOac(AO_ACno, _)),
not(production_step_output_AOac(AO_ACno, _)),
not(map_AOac_to_GLac(AO_ACno, GL_ACno)),
write("Can't map AOac to GLac for AOac "), write(AO_ACno), nl,
fail.

/* INDEXED_ASSERT_JEL_RTYPE--------------------------------------*/
indexed_assert_jel_Rtype(JEno, GL_ACno, Money_value, Money_date, Rtype);-
/* write(indexed_assert_jel_Rtype(JEno, GL_ACno, Money_value, Money_date, Rtype)),nl, */
price_level_accounting_switch(PLA_type),
( PLA_type = 'Historical Cost',
  assert_uncondit_jel(JEno, GL_ACno, Fno, UFno), /* asserts the FORMULA*/
  assertz(uncondit_formula(UFno, Money_value, [[]))
)

PLA_type = 'Current Purchasing Power',
get_price_index(g, Money_date, Gen_index_value),
( is_a(resource_hierarchy,Rtype,'Monetary Resource'), /* assert je for monetary account, UFno1 */
  assert_uncondit_jel(JEno, GL_ACno, Fno1, UFno1), /* Fno1 not used*/
  Coeff11 is Money_value,
  assertz(uncondit_formula(UFno1, Coeff11, [[]),
  /* now assert je for gearing gain, UFno2 */
  map_AC_concept_to_GLac('Gearing Gain', GG_GL_ACno),
  assert_uncondit_jel(JEno, GG_GL_ACno, Fno2, UFno2),
  Coeff21 is Money_value/Gen_index_value,
  assertz(uncondit_formula(UFno2, Coeff21, [[g,1]])),
  Coeff22 is -Money_value,
  assertz(uncondit_formula(UFno2, Coeff22, [[]))
)

; /*is_a(resource_hierarchy,Rtype,'Non-monetary (N.M.) Resource'), */
/* not gains or losses on non-monetary resource in CPP */
assert_uncondit_jel(JEno, GL_ACno, Fno3, UFno3),
Coeff31 is Money_value/Gen_index_value,
assertz(uncondit_formula(UFno3, Coeff31, [[g,1]]))
)

PLA_type = 'Current Cost Accounting',
resource_price_level_series_mapping(Rtype, Series),
get_price_index(g, Money_date, Gen_index_value),
get_price_index(Series, Money_date, Spec_index_value),
( is_a(resource_hierarchy,Rtype,'Monetary Resource'), /* assert je for monetary account, UFno1 */
  assert_uncondit_jel(JEno, GL_ACno, Fno1, UFno1), /* Fno1 not used*/
  Coeff11 is Money_value,
  assertz(uncondit_formula(UFno1, Coeff11, [[]),
  /* now assert je for gearing gain, UFno2 */
  map_AC_concept_to_GLac('Gearing Gain', GG_GL_ACno),
  assert_uncondit_jel(JEno, GG_GL_ACno, Fno2, UFno2),
  Coeff21 is Money_value/Gen_index_value,
  assertz(uncondit_formula(UFno2, Coeff21, [[g,1]])),
  Coeff22 is -Money_value,
  assertz(uncondit_formula(UFno2, Coeff22, [[]))
);
/* is_a(resource_hierarchy,Rtype,'Non-monetary (N.M.) Resource'), */
/* assert je for resource account, UFno1 */
assert_uncondit_jel(JEno,GL_ACno,Fno1,UFno1), /* Fno1 not used*/
Coeff1 is Money_value/Spec_index_value,
assertz(uncondit_formula(UFno1,Coeff11,[Series,1]));
/* now assert je for real gain above general prices, UFno2 */
(Series == g,
  true /* no Real Holding Gain, so no je required */
);  
map_AC_concept_to_GLac('Real Holding Gain',RHG_GL_ACno),
assert_uncondit_jel(JEno,RHG_GL_ACno,Fno2,UFno2),
Coeff21 is Money_value/Gen_index_value,
assertz(uncondit_formula(UFno2,Coeff21,[g,1])),
Coeff22 is -Coeff11,
assertz(uncondit_formula(UFno2,Coeff22,[Series,1]))

write('PLA_type is '),write(PLA_type),nl,
write('Error. assert jel Rtype not programmed for other than hc., csp., or '),
write('cca. accounting. Abort and correct.'),nl
).

/* END INDEXED_ASSERT_JEL_SERIES */

indexed_assert_jel_series(JEno,GL_ACno,Money_value,Money_date,Series):-
  write(indexed_assert_jel_Rtype(JEno,GL_ACno,Money_value,Money_date,Series)),nl,*
  price_level_accounting_switch(PLA_type),
  ( PLA_type = 'Historical Cost',
    assert_uncondit_jel(JEno,GL_ACno,Fno,UFno),
    assertz(uncondit_formula(UFno,Money_value,[]))
  ),
  PLA_type = 'Current Purchasing Power',
  get_price_index(g,Money_date,Gen_index_value),
  assert_uncondit_jel(JEno,GL_ACno,Fno3,UFno3),
  Coeff31 is Money_value/Gen_index_value,
  assertz(uncondit_formula(UFno3,Coeff31,[g,1]))
),
PLA_type = 'Current Cost Accounting',
  get_price_index(Series,Money_date,Spec_index_value),
  assert_uncondit_jel(JEno,GL_ACno,Fno1,UFno1), /* Fno1 not used*/
  Coeff1 is Money_value/Spec_index_value,
  assertz(uncondit_formula(UFno1,Coeff11,[Series,1]))
  /* now assert je for real gain above general prices, UFno2 */
  (Series == g,
    true /* no Real Holding Gain, so no je required */
  ),
  map_AC_concept_to_GLac('Real Holding Gain',RHG_GL_ACno),
  assert_uncondit_jel(JEno,RHG_GL_ACno,Fno2,UFno2),
  Coeff21 is Money_value/Gen_index_value,
  assertz(uncondit_formula(UFno2,Coeff21,[g,1])),
  Coeff22 is -Coeff11,
  assertz(uncondit_formula(UFno2,Coeff22,[Series,1]))
)

Interpreter - 13
write('PLA_type is '),write(PLA_type),nl,
write('Error: assert jel_series not programmed for other than hc., cpp., or ')
write('cca. accounting. Abort and correct.'),nl
).

/*---------------- GET PRICE INDEX------------------------*/
get_price_index(Index_series,Target_date,Index_value):-
date_function(D,Y,price_index(Index_series,D,Y),Target_date,Index_val),
/* date_function saves index values. To change indices, need to reinitialize */
( Index_val > 1,
  Index_value is Index_val
);
  Index_val =< 1, /* to stop nonsense jes */
  Index_value is 1
).

/*-------- INDEXEDASSERT_JEL_P_AND_L_ITEM-------------*/
indexed_assert_jel_P_and_L_item(JEno,GL_ACno,Money_value,Money_date):-
  price_level_accounting_switch(PLA_type),
  ( PLA_type = 'Historical Cost',
    assert_uncondit_jel(JEno,GL_ACno,Fno,UFno),
    assertz(uncondit_formula(UFno,Money_value[],[]))
  ;
  ( PLA_type = 'Current Purchasing Power',
    PLA_type = 'Current Cost Accounting'),
  /* no gearing gain on P&L items */
  get_price_index(g,Money_date,Gen_index_value),
  assert_uncondit_jel(JEno,GL_ACno,_,UFnol),
  Coeff is Money_value/Gen_index_value,
  assertz(uncondit_formula(UFnol,Coeff,[[]]))
  ;
  write('PLA_type is '),write(PLA_type),nl,
  write('Error: assert jel_P_L not programmed for other than hc., cpp., '),
  write('cca. accounting. Abort and correct.'),nl
).

/*------- DETERMINE MONEY VALUE FOR CORE JE LINE----------*/
determine_money_value_for_core_je_line(Eno,JEno,Date,AO_ACno,Q,V,Money_value,Money_date)
c:-
/* gets standard costs for manufacturing events, and identifies */
/* material price variances and labour rate variances for standard costing */
(fifo_lifo_switch(standard_cost),
ao_account(AO_ACno,_Class,Rno,Agent_id_),
write(ao_account(AO_ACno,_Class,Rno,Agent_id_),nl, *)
(unit_standard_cost(Rno,____)\derived unit_standard_cost(Rno,____)),
/* OK, have a standard cost for this resource*/
once(
  get_inflated_unit_standard_cost_of_resource(Rno,Date,
    Inflated_std_cost,Date_set),
  Money_value is Inflated_std_cost * Q,
  Money_date is Date
Above Written 1986/7, and checked carefully January 1991. OK!!

Note: The CPP/CCA value has been calculated by inflating the standard cost by the approp. price index. The value of inflating std cost is that the price variance is more meaningful, i.e., it is the best est. of the standard price on the day of purchase. After this time, CCA gains in inventory value (based on adjusted standard) over general prices will give rise to Relative Price Change CCA gains. So should the variance, if the use of standard costing is not to alter reported profit. For this reason, standard Money_value is returned at Date, not Date_set.

check if have variance from standard prices */
resource(Rno,_,Rtype),

once(
(Q>0),
    /* now have acquisition of resource with V determined by E.Event */
    /* so calculate price variance */
    Price_variance is V - Money_value, /*i.e., var at Date prices */
    ( Price_variance \= 0,
        ( is_a(resource_hierarchy,Rtype,'Unidentified Tangible Resources'),
          map_AC_concept_to_GLac('Materials Price Variance',GL_ACno)
        ;
        is_a(resource_hierarchy,Rtype,'Labour Services'),
        map_AC_concept_to_GLac('Labour Rate Variance',GL_ACno)
        ;
        write('Error: trying to post price variance in determine_money'),
        write({_value_for_core_je_line. Can't find mapping for Rtype'),
        nl,write(Rtype),nl
    ),
    indexed_assert_jel(Rtype,JEno,GL_ACno,Price_variance,
                      Date,Rtype)
    /* will get a CCA Gain above gen prices for CC/CPP */
    true /* i.e.,no price variance je required*/
) ;
true /* i.e., have standard cost, but Q<=0, or no V from EE */
)

( var(V),
    write('Error: no Money-value for event, fifo & lifo not yet coded for manuf.'),nl,
    fail
) ;

Money_value is V, /* this is the normal path for most E.Events*/
Money_date is Date
)

get_inflated_unit_standard_cost_of_resource
------------------------*/

gc(Rno):-gc(Rno,860501). */
gc(Rno,Event_date):- */
get_unit_standard_cost_of_resource(Rno,Event_date,Std_cost,Date_set), */

Interpreter - 15
/* write('Uninflated standard cost of resource '),write(Rno), */
/* write(' is '),write(Std_cost),nl. */

get_inflated_unit_standard_cost_of_resource(Rno,Event_date,Inflated_std_cost,
  Date_set):-
  get_unit_standardcost_of_resource(Rno,Event_date,Std_cost,Date_set),
  inflate_from_Date1_to_Date2_Rno(Date_set,Std_cost,Rno,Event_date,
    Inflated_std_cost).
/* write('Got inflated unit std cost of resource: '), */
/* write(Inflated_std_cost),write(' at Date '),write(Event_date),nl. */

/* WRITTEN 1986:
N.B. On what basis should "standard" costs be inflated?
Possible answers are: not at all,
by gen price index, and
by specific index.

Initially I chose the specific index, but found that with CCA the
various components credited in a interdept_transfer did not balance
the amount debited because each component cost was indexed at its
own rate, and the standard product cost was calculated at Date_set.

I then used the Inventory price index to inflate from Date_set to event date
for all standards for CCA, then the Gen price index from event date to Report
date, and each individual je balanced. However, the net balance of, say,
materials-in and materials-out will be non-zero (not checked).

E.g., Suppose std is set in month 1, goods are booked-in in month 5 and
out in month 7. Gen index rises linearly at 1% per month, Invent index
rises linearly at 2% per month. Std cost 1 Jan is $100. Report date is 31 Dec.
The May debit to MIP (in year end $) is 100 * (1+0.02*5)*(1+0.01*7)=117.7.
The July credit to MIP (in year end $) is 100*(1+0.02*7)*(1+0.01*5)=119.7.
Clearly the only way for debits to equal credits is to use the same index
throughout.

Using the Inventory index throughout for CCA, the debit is recorded in year-
end $ as 100*(1+0.02*12)=$124 with a Rel CCA gain for 7 months of $124-117.7.
The credit is $124, with a Rel CCA loss for 5 months of $124-119.7. The net
$2 gain is the amount the MIP appreciated in money-value over months 5 to 7.

It then seemed most correctly in the spirit of CCA to restate all the
component costs to Report date by their OWN indices and to recalculate the
standard cost of the output then. The debits and credits to, say, MIP, will
balance if calculated at one "standard", the Rel CCA gains will be calculated
on the component indices, and the je recording the transfer from one dept to
the next will balance because std cost of the product is the sum of the
inflated costs of the components. Thus storing computed costs of "products"
cannot be justified for CCA accounting.

THUS, AT PRESENT, COMPONENT STANDARD COSTS ARE INFLATED TO
TARGET_DATE BEFORE
THE "PRODUCT" STANDARD COST IS CALCULATED. THE REL CCA GAIN IS FROM
EVENT_DATE TO TARGET_DATE.
*/
get_unit_standard_cost_of_resource(Rno,Event_date,Std_cost,Date_set):-
    /* write("About to get unit std cost:");nl, */
    /* write(get_unit_standard_cost_of_resource(Rno,Event_date,Std_cost, */
    /* Date_set)),nl, */
    ( max(D_set,
        ( unit_standard_cost(Rno,D_set,_,_),
            Event_date >= D_set
        ),
        D_set),
    unit_standard_cost(Rno,Date_set,Std_cost)
    /* write("Uninflated standard cost for resource "),write(Rno), */
    /* write(' is '),write(Std_cost),nl */
    max(D_set,
        ( derived_unit_standard_cost(Rno,D_set,_,_,_),
            Event_date >= D_set
        ),
        D_set),
    derived_unit_standard_cost(Rno,Date_set,Std_cost,_,_,_)
    /* write("Uninflated derived standard cost for resource "),write(Rno), */
    /* write(' is '),write(Std_cost),nl */
    write("Unable to retrieve the std. cost of resource "),write(Rno),nl
).

/*-------- INFLATE_FROM_DATE1_TO_DATE2 PROCS ------------------------*/

inflate_from_Date1_to_Date2_Rno(Date1,Date1_money,Rno,Date2,Date2_money):-
    resource(Rno,Rno,Rtype),
    resource_price_level_series_mapping(Rtype,Series),
    inflate_from_Date1_to_Date2_series(Date1,Date1_money,Series,Date2,Date2_money).

inflate_from_Date1_to_Date2_Rtype(Date1,Date1_money,Rtype,Date2,Date2_money):-
    resource_price_level_series_mapping(Rtype,Series),
    inflate_from_Date1_to_Date2_series(Date1,Date1_money,Series,Date2,Date2_money).

inflate_from_Date1_to_Date2_series(Date1,Date1_money,Series,Date2,Date2_money):-
    price_level_accounting_switch(PLA_type),
    ( PLA_type = 'Historical Cost',
        Date2_money is Date1_money
    ;
    PLA_type = 'Current Purchasing Power',
        get_price_index(g,Date1,Index_then),
        get_price_index(g,Date2,Index_now),
        Date2_money is Date1_money * Index_now / Index_then
    ;
    PLA_type = 'Current Cost Accounting',
        get_price_index(Series,Date1,Index_then),
        get_price_index(Series,Date2,Index_now),
        Date2_money is Date1_money * Index_now / Index_then
    ).

/*---------------- COST OF GOODS ISSUED -----------------------------*/

/* NB. INCOMPLETE: */
At present, the program asserts inventory_shortfall(...). */
but this information is not used when new inventory is received. */
(A warning message will be written.)*/

The procedure cost_of_inventory_issued attempts to determine the lifo, fifo or standard cost of goods issued, either as COGS, or to a productive process.
Each acquisition of inventory has been recorded in a separate inventory_value record retrieved as follows:
retract(inventory_value(Rno,Eno,Date_set,Unit_cost,Qty_on_hand)),
and more than one record may need to be retracted, downgraded, and re-asserted if a large quantity is issued.

inventory_value(1028,1,870101,10,10).
inventory_value(1028,2,870201,11,10).
inventory_value(1028,3,870301,12,10).
inventory_value(1028,4,870401,13,10).
inventory_value(1028,5,870501,14,10).
fifo_lifo switch(lifo).
unit_standard_cost(1028,880101,100).
unit_standard_cost(1028,890101,110).
unit_standard_cost(1028,910101,120).
fifo_lifo switch(standard_cost).

vi(Q):-
get_cost_of_inventory_issued(1028,Q,900101,Qty_Date_Cost_list,Shortfall),
write('QDC_list ='),write(Qty_Date_Cost_list),nl,
write('Shortfall='),write(Shortfall),nl,
listing(inventory_value).

get_cost_of_inventory_issued(Rno,Q,Event_date,Qty_Date_Cost_list,Shortfall):-
( Q < 0,
calc_cost_and_update(Rno,Q,Event_date,Qty_Date_Cost_list,Shortfall) ;
write('Error. Q should be negative for all inventory issues. Q = '),
write(Q),nl).
calc_cost_and_update(Rno,Q,Event_date,Qty_Date_Cost_list,Shortfall):-
Qty_issued is -Q, /* Q, being an issue is -ve */
fifo_lifo switch(FLS),
( FLS == standard_cost,
get_unit_standard_cost_of_resource(Rno,Event_date,Std_cost,Date_set),
Qty_Date_Cost_list = [Qty_issued,Date_set,Std_cost],
Shortfall is 0
; (FLS == fifo; FLS == lifo),
calc_part_cost_and_update(FLS,Rno,Qty_issued,],Qty_Date_Cost_list,
Shortfall)
; write('No inventory_value info available for resource '),
write(Rno),nl).

Interpreter - 18
calc_part_cost_and_update(FLS,Rno,Qty_yet_to_downgrade,QDC_list_so_far,Final_QDC_list,Shortfall):-

  ( FLS == fifo, /* fifo uses earliest inventory first, i.e., min date */
    min(D_set,inventory_value(Rno,D_set,_,_),Date_set)
  ;
  FLS = lifo,
    max(D_set,inventory_value(Rno,D_set,_,_),Date_set)
  ;
  true /* std cost */
),

  retract(inventory_value(Rno,Eno,Date_set,Unit_cost,Qty_on_hand)),
  ( Qty_yet_to_downgrade =< Qty_on_hand,
    /* OK, have sufficient stock in this inventory record */
    append([[Qty_yet_to_downgrade,Date_set,Unit_cost]],
             QDC_list_so_far,Final_QDC_list),
    Qty_left is Qty_on_hand - Qty_yet_to_downgrade,
    Shortfall is 0,
    ( Qty_left = 0,
      asserta(inventory_value(Rno,Eno,Date_set,Unit_cost,Qty_left))
    ;
    true /* if inventory drops to zero, don't re-assert rec */
  )
  ;
  /* else need to get the next inventory_value record */
  Remain_Qty_yet_to_downgrade is Qty_yet_to_downgrade - Qty_on_hand,
  append([[Qty_on_hand,Date_set,Unit_cost]],
             QDC_list_so_far,Accum_QDC_list),
  calc_part_cost_and_update(FLS,Rno,Remain_Qty_yet_to_downgrade,
                           Accum_QDC_list,Final_QDC_list,Shortfall)
).

calc_part_cost_and_update(FLS,Rno,Qty_yet_to_downgrade,QDC_list_so_far,Final_QDC_list,Shortfall):-
  /* get here the "retract" at first line above must have failed */
  /* this means still looking for more stock - so must have a shortfall */
  write('Insufficient stock of Rno '),write(Rno),write(' recorded. Shortfall is '),nl,
  write('Qty_Date_Cost list so far is '),write(QDC_list_so_far),nl',
  Shortfall is Qty_yet_to_downgrade,
  Final_QDC_list = QDC_list_so_far.
/* ---------------------------------------------*/
/* DEPRECIATION AND AMORTIZATION ---------------*/
/* ---------------------------------------------*/

td:-
td(850101),
td(Date):-
  retractall(formula(_,_,_,_,_,_,)),
  retractall(uncondit_formula(_,_,_,_,_,_,)),
  generate_acq_depr_amort_jets(1,800101,12402,1,36500,DFno,CFno),
  list_formula(DFno,Date,Value_of_depr_expense),
  list_formula(CFno,Date,Value_of_accum_depr).

/*expected_value(1103,uniform,120,10).* /
/*expected_value(1103,'Reducing balance',0,10).* /
/*expected_value(1103,'Reducing balance',120,10). */

/* GENERATE DEPRECIATION & AMORTIZATION JELS (ON ACQISITION) */

/* first deal with case where life is <= zero, e.g., phone bill */

generate_acq_depr_amortjels(JEno,Date,AO_ACno,Q,Acquisition_cost,DFno,CFno):-
    /* write(generate_acq_depr_amortjels(JEno,Date,AO_ACno,Q,Acquisition_cost, */
    /* DFno,CFno)),nl, */
    ao_account(AO_ACno,Class,Rno,Agent_id_1),
    resource(Rno,_,Rtype),
    expected_value(Rno,Expected_pattern_of_benefits,
        Expected_life_in_months,Expected_residual_value_per_cent),
    Expected_life_in_months @<= 0,
    write('Zero life shrinking value asset, expensing immediately'),nl,
    map_AOac_to_GLac(AO_ACno,Gl_ACno),
    ( is_a(resource_ev_hierarchy,Rtype,'Tangible Resource'),
        gl_depreciation(GL_ACno,Cr_ACno,Dr_ACno),
        accum_depr_ac_for_asset(GL_ACno,Cr_ACno),
        expense_immediately(JEno,Date,AO_ACno,Q,
            Acquisition_cost,Dr_ACno,Cr_ACno,DFno,CFno)
    ;
    is_a(resource_ev_hierarchy,Rtype,'Intangible Resource'),
    gl_amortization(GL_ACno,Dr_ACno),
    expense_immediately(JEno,Date,AO_ACno,Q,
        Acquisition_cost,Dr_ACno,Gl_ACno,DFno,CFno)
    /* note Cr account above is the asset itself */
    ;
    write('Error: invalid Rtype for depr amort, Rtype='),write(Rtype),nl
).

generate_acq_depr_amortjels(JEno,Date,AO_ACno,Q,Acquisition_cost,DFno,CFno):-
    /* write(generate_acq_depr_amortjels(JEno,Date,AO_ACno,Q,Acquisition_cost, */
    /* DFno,CFno)),nl, */
    ao_account(AO_ACno,Class,Rno,Agent_id_),
    resource(Rno,_,Rtype),
    expected_value(Rno,Expected_pattern_of_benefits,
        Expected_life_in_months,Expected_residual_value_per_cent),
    Expected_life_in_months @>= 0,
    /* the normal case */
    map_AOac_to_GLac(AO_ACno,Gl_ACno),
    ( is_a(resource_ev_hierarchy,Rtype,'Tangible Resource'),
        ( Expected_pattern_of_benefits == uniform;
            /* Expected_pattern_of_benefits == 'Straight line'),
            gl_depreciation(GL_ACno,Cr_ACno,Dr_ACno),
            accum_depr_ac_for_asset(GL_ACno,Cr_ACno),
            generate_str_line_deprjels(JEno,Date,AO_ACno,Q,
                Acquisition_cost,Dr_ACno,Cr_ACno,DFno,CFno)
        ;
        Expected_pattern_of_benefits == 'Reducing balance',
        gl_depreciation(GL_ACno,Cr_ACno,Dr_ACno),
        accum_depr_ac_for_asset(GL_ACno,Cr_ACno),
    /* */
        )
).
generate_red_bal_depr_jels(JEno, Date, AO_ACno, Q, Acquisition_cost, Dr_ACno, Cr_ACno, DFno, CFno),
true
)

is_a(resource_ev_hierarchy, Rtype, 'Intangible Resource'),
gl_amortization(GL_ACno, Dr_ACno),
generate_st_line_depr_jels(JEno, Date, AO_ACno, Q, Acquisition_cost, Dr_ACno, GL_ACno, DFno, CFno)
/* note Cr account above is the asset itself */

write('Error2 invalid Rtype for depr_amort, Rtype=', Rtype), nl
true.

/* ——— expense immediately ————————*/
expense_immediately(JEno, Date, AO_ACno, Q, Acquisition_cost, Dr_ACno, Cr_ACno, DFno, CFno):-
indexed_assert_jel_P_and_L_item(JEno, Dr_ACno, Acquisition_cost, Date),
Cr_acquisition_cost is -Acquisition_cost,
ao_account(AO_ACno, Class, Rno, Agent_id_),
resource(Rno, Unit_of_measure, Rtype),
indexed_assert_jel_Rtype(JEno, Cr_ACno, Cr_acquisition_cost, Date, Rtype).

/* ——— straight line depreciation AND amortization ————*/
generate_st_line_depr_jels(JEno, Date, AO_ACno, Q, Acquisition_cost, Dr_ACno, Cr_ACno, DFno, CFno):-
/* write(generate_st_line_depr_jels(JEno, Date, AO_ACno, Q, */
/* Acquisition_cost, Dr_ACno, Cr_ACno, DFno, CFno)), nl, */
ao_account(AO_ACno, Class, Rno, Agent_id_),
resource(Rno, Unit_of_measure, Rtype),
expected_value(Rno, Expected_pattern_of_benefits, Expected_life_in_months, Expected_residual_value_per_cent),
map_AOac_to_GLac(AO_ACno, GL_ACno),

/* if(d-d0)<(Acq/Daily_rate), Daily_depr_amount*(d-d0), Acq_cost */
write('Starting straight line depreciation for Rno=', Rno), nl,
Expected_life_in_days is Expected_life_in_months * 30.4375,
Daily_depr_amount is Acquisition_cost * (1 - Expected_residual_value_per_cent /100)/(Expected_life_in_days),
Max_days_to_depr is Acquisition_cost/Daily_depr_amount,
reference_date_for_formulai(Ref_date),
number_of_days_from_to(Ref_date, Date, Day0),
Minus_Day0 is -Day0,
/* Dr Depr expense */
Condit='<',
assert_condit_jel(JEno, Dr_ACno, DFno, LHS_UFno, Condit, RHS_UFno, True_UFno, False_UFno),
assertz(uncondit_formula(LHS_UFno, Minus_Day0, [], [])),
assertz(uncondit_formula(LHS_UFno, 1, [[d, 1]])),
assertz(uncondit_formula(RHS_UFno, Minus_Day0, Max_days_to_depr, [])),
Coeff is Daily_depr_amount*Minus_Day0,
price_level_accounting_switch(PLA_type),
( PLA_type = 'Historical Cost',
  Const_term = [],
  Days_term = [[d,1]],
  DE_Coeff is Coeff,
  DE_Daily_depr_amount is Daily_depr_amount,
  DE_Acquisition_cost is Acquisition_cost
);

( PLA_type = 'Current Purchasing Power',
  get_price_index(g,Date,Gen_index_value),
  Const_term = [[g,1]],
  Days_term = [[d,1],[g,1]],
  DE_Coeff is Coeff/Gen_index_value,
  DE_Daily_depr_amount is Daily_depr_amount/Gen_index_value,
  DE_Acquisition_cost is Acquisition_cost/Gen_index_value
);

( PLA_type = 'Current Cost Accounting',
  resource_price_level_series_mapping(Rtype,Series),
  get_price_index(Series,Date,Spec_index_value),
  Const_term = [[Series,1]],
  Days_term = [[d,1],[Series,1]],
  DE_Coeff is Coeff/Spec_index_value,
  DE_Daily_depr_amount is Daily_depr_amount/Spec_index_value,
  DE_Acquisition_cost is Acquisition_cost/Spec_index_value
),

assertz(uncondit_formula(True_UFno,DE_Coeff,Const_term)),
assertz(uncondit_formula(True_UFno,DE_Daily_depr_amount,Days_term)),
assertz(uncondit_formula(False_UFno,DE_Acquisition_cost,Const_term)),
/* Ct Accum Depr */
assertz(uncondit_formula(TEno,Cr_ACho,CFno,CLHS_UFno,Condit,CRHS_UFno,
  CTrue_UFno,CFFalse_UFno),
assertz(uncondit_formula(CLHS_UFno,MinusDay0,[],)),
assertz(uncondit_formula(CLHS_UFno,Max_days_to_depr,[],)),
Minus_Coeff is -Coeff,
Minus_Daily_depr_amount is -Daily_depr_amount,
Minus_Acquisition_cost is -Acquisition_cost,
price_level_accounting_switch(PLA_type),

( PLA_type = 'Historical Cost',
  AD_Coeff is Minus_Coeff,
  AD_Daily_depr_amount is Minus_Daily_depr_amount,
  AD_Acquisition_cost is Minus_Acquisition_cost
);

( PLA_type = 'Current Purchasing Power',
  AD_Coeff is Minus_Coeff/Gen_index_value,
  AD_Daily_depr_amount is Minus_Daily_depr_amount/Gen_index_value,
  AD_Acquisition_cost is Minus_Acquisition_cost/Gen_index_value
);

( PLA_type = 'Current Cost Accounting',
  AD_Coeff is Minus_Coeff/Spec_index_value,
  AD_Daily_depr_amount is Minus_Daily_depr_amount/Spec_index_value,
  AD_Acquisition_cost is Minus_Acquisition_cost/Spec_index_value
),

assertz(uncondit_formula(CTrue_UFno,AD_Coeff,Const_term)),
assertz(uncondit_formula(CTrue_UFno,AD_Daily_depr_amount,Days_term)),
assertz(uncondit_formula(CFalse_UFno,AD_Acquisition_cost,Const_term)),
/* listing(uncondit_formula),*/

Interpreter - 22
true.

/ * --- reducing balance depreciation --- */

generate_red_bal_depr_jel(JEno,Date,AO_ACno,Q,
Acquisition_cost,Dr_ACno,Cr_ACno,DFno,CFno):-

/*
  V = Acq_cost * Reduction_factor^Days_since_acq */
/*
at end, V_end = Acq_cost * Reduction_factor^Life_in_days */
/*
  therefore, Reduction_factor = (V_end / Acq_cost)^(1/Life_in_days) */
/*
  therefore, Amort = Acq_cost*(1-Reduction_factor^Days_since_acq) */
/*
  therefore, Amort = Acq_cost*(1-Resid%/100^((Days_since_acq/Life_in_days))) */

ao_account(AO_ACno,_,Class,Rno,Agent_id,,,
resource(Rno,_,_,Rtype),
expected_value(Rno,Expected_pattern_of_benefits,
  Expected_life_in_months,Expected_residual_value_per_cent),
map_AOac_to_GL_ac(AO_ACno,GL ACno),

write('Starting reducing bal depreciation code for Rno='),write(Rno),nl,
/* first assert exponential term, it is independent of PLA switch */
/*
  use this exponent for both Dr Depr Exp, and Cr Accum Depr */
get_next_UFno(Exp_UFno),
reference_date_for_formulae(Ref_date),
number_of_days_from_to(Ref_date,Date,DayO),
Expected_life_in_days is Expected_life_in_months * 30.4375,
Exp_Coeff1 is -DayO/Expected_life_in_days,
Exp_Coeff2 is 1/Expected_life_in_days,
assertz(uncondit_formula(Exp_UFno,Exp_Coeff1,[])),
assertz(uncondit_formula(Exp_UFno,Exp_Coeff2,[[d,1]])),
/*
  write('Formula for exponent is:'),nl,
  */
write(assert_uncondit_jel(JEno,Dr ACno,DFno,DE_UFno)),nl,
/*
  Dr Depr expense */
write(assert_uncondit_jel(JEno,Dr ACno,DFno,DE_UFno)),nl,
assert_uncondit_jel(JEno,Dr ACno,DFno,DE_UFno),
Residual_fraction is Expected_residual_value_per_cent/100,
price_level_accounting_switch(PLA_type),
( PLA_type = 'Historical Cost',
  DE_term1 = [],
  DE_term2 = [[Residual_fraction,[Exp_UFno]]],
  DE_Coeff1 is Acquisition_cost,
  DE_Coeff2 is -Acquisition_cost
)

/*
  PLA_type = 'Current Purchasing Power',
  get_price_index(g,Date,Gen_index_value),
  DE_term1 = [[g,1]],
  DE_term2 = [[g,1],[Residual_fraction,[Exp_UFno]]],
  DE_Coeff1 is Acquisition_cost/Gen_index_value,
  DE_Coeff2 is -Acquisition_cost/Gen_index_value
)

/*
  PLA_type = 'Current Cost Accounting',
  resource_price_level_series_mapping(Rtype,Series),
  get_price_index(Series,Date,Spec_index_value),
  DE_term1 = [[Series,1]],

Interpreter - 23
DE_term2 = [[Series,1],[Residual_fraction,[Exp_UFno]]],
DE_Coeff1 is Acquisition_cost/Spec_index_value,
DE_Coeff2 is -Acquisition_cost/Spec_index_value,
assertz(uncondit_formula(DE_UFno,DE_Coeff1,DE_term1)),
assertz(uncondit_formula(DE_UFno,DE_Coeff2,DE_term2)),
/* write('Formula for base of Debit red bal formula is:'),nl, */
/* write(uncondit_formula(DE_UFno,DE_Coeff1,DE_term1))),nl, */
/* write(uncondit_formula(DE_UFno,DE_Coeff2,DE_term2))),nl, */

/* Cr Accum Depr */
write(assert_uncondit_jel(JEno,Cr_ACno,CFno,AC_UFno)),nl, */
( PLA_type = 'Historical Cost',
AC_Coeff1 is -Acquisition_cost,
AC_Coeff2 is Acquisition_cost
);
PLA_type = 'Current Purchasing Power',
AC_Coeff1 is -Acquisition_cost/Gen_index_value,
AC_Coeff2 is Acquisition_cost/Gen_index_value
);
PLA_type = 'Current Cost Accounting',
AC_Coeff1 is -Acquisition_cost/Spec_index_value,
AC_Coeff2 is Acquisition_cost/Spec_index_value
),
assertz(uncondit_formula(AC_UFno,AC_Coeff1,DE_term1)),
assertz(uncondit_formula(AC_UFno,AC_Coeff2,DE_term2)),
/* write('Formula for base of Credit red bal formula is:'),nl, */
/* write(uncondit_formula(AC_UFno,AC_Coeff1,DE_term1))),nl, */
/* write(uncondit_formula(AC_UFno,AC_Coeff2,DE_term2))),nl, */
/* note DE_terms 1 and 2, not AC_ above*/
listing(uncondit_formula) /*
write('got to here OK'),nl, */
true.

/* NOT YET CONVERTED TO FORMULA ACCOUNTING
sum of years digits depreciation ---------------
Expected_pattern_of_benefits = 'Sum of years digits',
Whole_years_life is Expected_life_in_months // 12,
Remaining_months_life is Expected_life_in_months mod 12,
( Remaining_months_life > 0,
Life_in_years is Whole_years_life + 1
);
Life_in_years is Whole_years_life
)
sum_of_years_digits(Life_in_years,SOYD),
Years_till_now is Days_till_now // 365,
Remaining_days_till_now is Days_till_now mod 365,
sum_of_weights_till_now is Days_till_now,Sum_of_weights_till_now),
write('sum of weights'),write(Sum_of_weights_till_now),nl,
Sum_of_annual_depr is Acquisition_cost * (1 - Expected_residual_value_percent / 100 ) *
Sum_of_weights_till_now / SOYD,
Annual_depr_amount is Acquisition_cost * (1 - Expected_residual_value_percent / 100 ) *
(Life_in_years - Years_till_now) / SOYD,
Daily_depr_amount is Annual_depr_amount /365.25,
Amort is Sum of annual depr +
   Daily depr_amount * Remaining days till now
;
write('not yet coded for other than uniform or SOYD shrinking'),nl,
Amort = 0
);
( Amort =< Acquisition_cost,
   Amort_amount is Amort
);
( Amort > Acquisition_cost,
   Amort_amount is Acquisition_cost
);
write('Amort_amount from '),write(Acquisition_date),write(' to '),
write(Date_now),write(' is '),write(Amort_amount),nl.

sum_of_years_digits(0,0).
sum_of_years_digits(Life in years,SOYD):-
  L is Life in years - 1,
  sum_of_years_digits(L,S),
  SOYD is S + Life in years.
*/*/  
\*\* |L,N\*:\*sum of weights till now(L,N,S),write(S),nl. */\*\*  
\*\* (9,0)\*\0\*:\*0, (9,1)\*:\*9, (9,2)\*\*:\*9+8\*:\*17, (9,9)\*:\*45, (9,20)\*:\*45. */\*\*  

sum_of_weights_till_now(Life in years,0,0).
sum_of_weights_till_now(Life in years,Years_till_now,Sum of weights till now):-
  ( Years_till_now < Life in years,
    Y is Years_till_now - 1
  );
  Years_till_now >= Life in years, /* don't overdepreciate */
  Y is Life in years - 1
),
sum_of_weights_till_now(Life in years,Y,S),
Sum of weights till now is S + (Life in years - Y).

/* Manufacturing stuff */

/* --------------------- CALC DERIVED STANDARD COSTS ---------------------*/
calc_derived_standard_costs:-
  /* write('About to start calc_derived_standard_costs'),nl, */
  (retractall(derived_unit_standard_cost(_,_,_,_,_,_));true),
  fast_setof(Date_set,unit_standard_cost(_,Date_set,J,Standard dates),
  fast_setof(Rno,composite_resource(Rno,_,_),Derivable cost resources),
  write('Dates on which standards have been set are: '),
  write(Standard dates),nl,
  write('Rnos of resources for which standard costs are to be derived are: '),
  write(Derivable cost resources),nl,
  calculate_all_standard_costs(Standard dates,Derivable cost resources),
  listing(derived_unit_standard_cost).
calc_derived_standard_costs:-

Interpreter - 25
write("No standard costs derivable").nl.

calculate_all_standard_costs([],_).
calculate_all_standard_costs([Head_date, Remaining_dates], Derivable_cost_resources).
calculate_all_standard_costs_set_at_one_date(Head_date, Derivable_cost_resources).
calculate_all_standard_costs(Remaining_dates, Derivable_cost_resources).

calculate_all_standard_costs_set_at_one_date([], []).
calculate_all_standard_costs_set_at_one_date(Standard_date, [Head_Rno | Remaining_Rnos]).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

calculate_unit_standard_cost_of_resource(Head_Rno, Standard_date, _).

/* This proc was written to calculate the unit standard_cost of intermediate and final products from the basic ingredients. It uses Standard_date to select the appropriate ingredient standard cost (different standards for one ingredient at different times), then calculates the uninflated product standard cost at that date. */

max(D_set,  
  (  composite_resource(Rno, D_set, _),  
    Standard_date @>= D_set  
  ),  
  Date_defined),  

/* Date_defined is the date the physical structure of the composite was defined. It will change less frequently than costs */

composite_resource(Rno, Date_defined, Component_Rno, Qty_of_component),

once((
  resource(Component_Rno, __, Rtype),  
  (  is_a(resource_ev_hierarchy, Rtype, 'Inventory'),  
    get_unit_standard_cost_of_resource(Component_Rno,  
      Standard_date, Component_std_cost, Date_set),  
    /* above assumes all standards set simultaneously, else */  
    /* need to inflate from its Date_set to this Date_set */  
    Q_by_C is Qty_of_component * Component_std_cost,  
    assert(uninflated_Q_by_C_of_component(Rno, material_cost,  
      Q_by_C, Date_set))  
  ),  
  (  is_a(resource_ev_hierarchy, Rtype, 'Labour Services'),  
    get_unit_standard_cost_of_resource(Component_Rno, Standard_date,  
      Component_std_cost, Date_set),  
    Q_by_C is Qty_of_component * Component_std_cost,  
    assert(uninflated_Q_by_C_of_component(Rno, labour_cost,  
      Q_by_C, Date_set)),  
  );  

/* also need to calculate OH: */
department_responsible_for_manufacture(Rno,Department),
get_departmental_OH_rate(Department,'dlh',Std_date,

OH_rate,Date_set),
/*
write(get_departmental_OH_rate(Department,'dlh',Std_date, */

/*
OH_rate,Date_set)),nl, */
Unit_OH is OH_rate \* Qty_of_component,
assert(uninflated_Q_by_C_of_component(Rno,overhead_cost,

Unit_OH,Date_set))
true /* ignore machine hours */
fail.
calculate_unit_standard_cost_of_resource(Rno,Std_date,Std_cost,

M_std_cost,L_std_cost,OH_std_cost):-
sum(Q_by_C,

uninflated_Q_by_C_of_component(Rno,material_cost,Q_by_C,),
M_std_cost),

sum(Q_by_C,

uninflated_Q_by_C_of_component(Rno,labour_cost,Q_by_C,),
L_std_cost),

/* need to add setup costs to overhead cost */

sum(Q_by_C,

uninflated_Q_by_C_of_component(Rno,overhead_cost,Q_by_C,),
OH_std_cost),

Std_cost is M_std_cost+L_std_cost+OH_std_cost,
retractall(uninflated_Q_by_C_of_component(Rno,_,_,_)).

/*——— GET DEPARTMENTAL OH RATE ————————————*/

get_inflated_departmental_OH_rate(Department,Basis_of_variation,Event_date,

Inflated_OH_rate):-
get_departmental_OH_rate(Department,Basis_of_variation,Event_date,

OH_rate,Date_set),
/* inflate with Plant index since depr is a major part of OH */
inflate_from_Date1_to_Date2_series(Date_set,OH_rate,

'Plant Price Index',Event_date,Inflated_OH_rate).

get_departmental_OH_rate(Department,Basis_of_variation,Event_date,

OH_rate,Date_set):-
(max(D_set,

departmental_OH_rate(Department,D_set,Basis_of_variation),

Event_date >= D_set
),

Date_set),
departmental_OH_rate(Department,Date_set,OH_rate,Basis_of_variation)
/*write("Latest overhead rate for department "),write(Department), */
/*write(" is "),write(OH_rate),nl */
;
write("Error: no current OH_rate and/or Basis of variation for department "),
write(Department),nl,
OH_rate is 0,
write("OH rate is set to zero."),nl
).
/*--- GET DEPARTMENT OH FLEXIBLE BUDGET ---------------------------*/

get_inflated_department_OH_flexible_budget(Department,Event_date, Inflated_fixed_OH,Inflated_var_OH_coefficient,Basis_of_variation):-
get_department_OH_flexible_budget(Department,Event_date,Date_set,Fixed_OH, Var_OH_coefficient,Basis_of_variation),
/* inflate with Plant index since depr is a major part of OH */
inflate_from_Date1_to_Date2_series(Date_set,Fixed_OH, 'Plant Price Index',Event_date,Inflated_fixed_OH),
inflate_from_Date1_to_Date2_series(Date_set,Var_OH_coefficient, 'Plant Price Index',Event_date,Inflated_var_OH_coefficient).

get_department_OH_flexible_budget(Department,Event_date,Date_set,Fixed_OH, Var_OH_coefficient,Basis_of_variation):-
( max(D_set,
  ( department_OH_flexible_budget(Department,D_set,_,_,Basis_of_variation),
    Event_date > = D_set
  ),
  Date_set,
  department_OH_flexible_budget(Department,Date_set,Fixed_OH, Var_OH_coefficient,Basis_of_variation)
)
/*write('Latest fixed overhead for department '),write(Department), */
/*write(' is '),write(Fixed_OH),nl, */
/*write('Latest variable overhead coeff. for department '),write(Department), */
/*write(' is '),write(Var_OH_coefficient),nl */
write('Error: no current Fixed_OH/Var_OH_coefficient and/or Basis_of_variation for department '),
write(Department),nl,
Fixed_OH is 0,
Var_OH_coefficient is 0,
write('Flexible OH variables are set to zero.'),nl
).

/*------------------- TRASH -----------------------------------*/

inflate_jes(New_report_date):-
/* inflates jes generated for a given report date to a new date, to simplify comparisons of opening balances of later periods */

once( ( report_date(Report_date),
inflate_from_Date1_to_Date2_series(Report_date,1,g, New_report_date,Inflation_factor)
);
Inflation_factor = 1
).

(start_accounting_period(Start_accounting_period);true),
write('INFLATED Journal entries for REEFA LTD, for period from '),
write(Start_accounting_period),nl,write(' to Report date '),write(Report_date),
write(' IN DOLLARS OF '),write(New_report_date),nl,
price_level_accounting_switch(S1),
filo_lifo_switch(S2),
leased_assets_on_off_balance_sheet(S3),

Interpreter - 28
write('for '),write(S1),write(', '),write(S2),write(', and leased assets '),
write(S3),write(' balance sheet.'),nl,nl

jeh(JEno,Date,Eno,Narrative),
once(list_inflated_je(JEno,Inflation_factor)),
fail.

inflate_jes(New_report_date).

list_inflated_je(JEno,Inflation_factor):-
    jeh(JEno,Date,Eno,Narrative),
    write(jeh(JEno,Date,Eno,Narrative)),nl,
    list_inflated_jeds(JEno,Inflation_factor),nl.

list_inflated_jeds(JEno,Inflation_factor):-
    jed(JEno,L,GL_ACno,Report_amount),
    once((
        Amount is Report_amount • Inflation_factor,
        write('),write(JEno),write(', '),
        write(L),write(', '),
        write(GL_ACno),write(' ')
    
    chart_of_accounts(GL_ACno,Description,Class),
    write('<'),write fixed length(Description,35),write('>, ')
    
    write(')'),write_fixed_length(Description,35),write(')'),nl
    
    )
fail.

list_inflated_jeds(JEno).
assert_core_jel_line(Eno,Lno,JEno,Date,ACno,Q,V) .......................... 10
assert_jel(JEno,Date,Eno,Narrative) ........................................... 10
assert_jel via AO_mapping(JEno,ACno,Money_value,Money_date) .......... 11, 12
calc_cost_and_update(Rno,Q,Event_date,Qty_Date,Cost_list,Short) .... 18
calc_derived_standard_costs ...................................................... 25
calc_part_cost_and_update(FLS,Rno,Qty_yet_to_downgrade,QDC_list) .... 19
calc_unit_standard_cost_of_resource(Rno,Standard_date,Std_cost) ..... 26
calculate_all_standard_costs([Head_date|Remaining_dates],Dev) .... 26
calculate_all_standard_costs_set_at_one_date(Standard_date) ...... 26, 27
calculate_unit_standard_cost_of_resource(Rno,Standard_date,Std) .. 26, 27
calculate ................................................................. 25
compare_jel(JEno) ............................................................. 9
determine_money_value_for_core_jel_line(Eno,JEno,Date,ACno,Q .... 14
expense_immediately(JEno,Date,ACno,Money_value) .................. 21
filter_out_conditional_assets_and_obligations(ACno) .................. 11
generate_seq_depr_amort_jel(JEno,Date,ACno,Acquisition_c ...... 20
generate_jel Eds .................................................................. 7, 8
generate_red_bal_depr_jel(JEno,Date,ACno,Q) ......................... 23
generate_unit_depr_jel(JEno,Date,ACno,Q) ................................. 18
generate ................................................................. 18
get_cost_of_inventory_issued(Rno,Q,Event_date,Qty_Date,Cost_list) .. 28
generate_departmental_OH_rate(Department,Event_date,Date) .... 28
generate_inflated_departmental_OH_rate(Department,Event_date,Date) .... 27
get_inflated_unit_standard_cost_of_resource(Rno,Event_date,Inf) ... 27
get_price_index(Index_series,Target_date,Index_value) .............. 14
generate ................................................................. 14
indexed_assert_jel_P_and_L_item(JEno,GL_ACno,Money_value,Money .... 12
indexed_assert_jel_Rtype(JEno,GL_ACno,Money_value,Money_date,R .... 10
indexed ................................................................. 10
inflated_depr_from_Date1_to_Date2(Rno,Date1,Date2,money) .......... 17
inflated_depr_from_Date1_to_Date2(Rtype,Date1,Date2,money) .......... 17
inflated_depr_from_Date1_to_Date2_series(Series,Date1,Date2,money,Series) .... 17
inflated_jel(New_report_date) .................................................. 28
initialize_model(Start_accounting_period,Report_date) ............... 3
invalid_money_value(Money_value) ............................................ 11
list_inflated_jel(JEno,Inflation_factor) ..................................... 29
list_inflated_jelds(JEno,Inflation_factor) ................................. 29
initialize ................................................................. 5
loop through_event_detail_lines(JEno,Eno,Date) .......................... 10
multiview(Start_accounting_period,Report_date) ....................... 2
mv ................................................................. 2
mv(Start_accounting_period,Report_date) ...................................... 2
process_event(Eno,Date,Etype,Docno,AgentID,Narrative,J .... 3
process_events_for_period(Start_period_date,Next_closing_date) ... 3
ram ................................................................. 4
red ................................................................. 5
riv ................................................................. 5
set_fifo_lifo_switch .......................................................... 6
set_leased_assets_on_off_balance_sheet_switch ......................... 7
set_price_level_accounting_switch ......................................... 6
set_up_closing_dates(Start_period_date,Report_date) .................. 5
set_up_dates(Start_accounting_period,Report_date) ................... 5
Interpreter - 30
set_variance_analysis_frequency .......................... 7
simple(Eno,Date,Etype,Doctype,Doceno,AgentID,Narrative,JEno) .......................... 10
sum_of_weights_till_now(Life_in_years,Years_till_now,Sum_of_we .......................... 25
sum_of_years_digits(Life_in_years,SOYD) .......................... 25
td  ........................................................................ 19
td(Date)  ................................................................ 19
value_res(Eno,Date,Etype) ....................................... 8, 9
\nu(Q)  ................................................................. 18
APPENDIX 3: SPECIAL EVENT HANDLERS FOR INTERPRETER

This Appendix contains 504 lines (10 pages) of Prolog code special event handlers for the Interpreter. These procedures are as follows:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>cash_payment</td>
<td>2</td>
</tr>
<tr>
<td>cash_receipt</td>
<td>2</td>
</tr>
<tr>
<td>credit_sale</td>
<td>2</td>
</tr>
<tr>
<td>donation</td>
<td>5</td>
</tr>
<tr>
<td>issue_shares</td>
<td>6</td>
</tr>
<tr>
<td>manuf_event</td>
<td>8</td>
</tr>
<tr>
<td>open_acs</td>
<td>9</td>
</tr>
<tr>
<td>production_summary</td>
<td>10</td>
</tr>
<tr>
<td>sale_depreciable_asset</td>
<td>10</td>
</tr>
</tbody>
</table>

These procedures are invoked by line 84 in the Interpreter in its main loop. Basically, event headers are retrieved one-by-one on line 59 of the Interpreter. If the event date is in the relevant period, line 64 invokes process_event. On line 75 of the Interpreter, process_event determines which event handler is to be used for this event type, and the appropriate procedure is invoked by line 84. The event handler called simple is in the Interpreter. All other event handlers are listed in this Appendix.

There is an index to all procedures in this Appendix on page 12.
/* ------------------ CASH PAYMENT ------------------ */
cash_payment(Eno,Date,Event,Doctype,Docno,AgentID,Narrative,JEno):-
  write('cash_payment called for event '),write(Eno),nl,
  assert_je(JEno,Date,Eno,Narrative),
  loop_through_event_detail_lines(JEno,Eno,Date), /* always true */ !,
  /* The balancing item for all jes so far must be discount revenue */
  check_je_balance(JEno,'Balance formulae required'),
  once((fast_setof(Coeff,balance_formula_regd(_,_,Coeff,_,_,J,Coeffs),
    /* write('Coeffs for balance formula ='),write(Coeffs),nl, */
    ( Coeffs==[0]
    ;
    map_AC_concept_to_GLac('Discount Revenue',GL_ACno),
    assert_balancing_je_lines(JEno,GL_ACno)
    )
  )).

/* ------------------ CASH RECEIPT ------------------ */
cash_receipt(Eno,Date,Event,Doctype,Docno,AgentID,Narrative,JEno):-
  write('cash receipt called for event '),write(Eno),nl,
  assert_je(JEno,Date,Eno,Narrative),
  loop_through_event_detail_lines(JEno,Eno,Date), /*always true */ !,
  /* The balancing item for all jes so far must be discount expense */
  check_je_balance(JEno,'Balance formulae required'),
  fast_setof(Coeff,balance_formula_regd(_,_,Coeff,_,_,J,Coeffs),
    /* write('Coeffs for balance formula ='),write(Coeffs),nl, */
    ( Coeffs==[0]
    ;
    map_AC_concept_to_GLac('Discount Expense',GL_ACno),
    assert_balancing_je_lines(JEno,GL_ACno)
    )
  ).

/* ------------------ CREDIT SALE ------------------ */
credit_sale(Eno,Date,Event,Doctype,Docno,AgentID,Narrative,JEno):-
  write('credit sale called for event '),write(Eno),nl,
  assert_je(JEno,Date,Eno,Narrative), /* from Interpreter core */ !,
  /* Reminder:need to provide for doubtful debts: 
    Dr Accounts Receivable 100
    Cr Sales 100
    Dr Doubtful debts exp 2.5
    Cr Allowance for DDs 2.5
    Actually find 2.5% is too high. 1.6% is better for data given */
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
  !,
assert_credit_sale_line(JEno,AO_ACno,Q,V,Date):-
    once((filter_out_conditional_assets_and_obligations(AO_ACno)),
        /* Dr A/C Receivable, Cr Sales Revenue; V computed in generate_eds */
        assert_credit_sales_line(JEno,AO_ACno,Q,V,Date),
        map_AC_concept_to_GLac('Sales Revenue',GL_ACno),
        Money_value is -V,
        indexed_assert jel_P_and_L_item(JEno,GL_ACno,Money_value,Date)
    ;
    Q == 0 /* do nothing */
).

fail.

assert_jets_for_inventory(JEno,AO_ACno,Date,[],Shortfall).
assert_jets_for_inventory(JEno,AO_ACno,Date,[H | T],Shortfall):-
    [Qty,Date_acq,Unit_cost]=H,
    /* Coded so Q_D_U valid for standard cost, but Shortfall always = 0 */
    Shortfall \= 0,
    fifo_lifo_switch(FLS),
    ( FLS == fifo,
      Money_value is (Qty + Shortfall) * Unit_cost,
      Short is 0
    ;
    FLS == lifo,
    T == [], /* use latest date cost for valuing Shortfall */
    Money_value is (Qty + Shortfall) * Unit_cost,
    Event_handler - 3
Short is 0

; Short is Shortfall /* lifo, but not tail of list */

; Money_value is Qty * Unit_cost,

/* Dr CoGS, Cr Inventory */

map_AC_concept_to_GLac('Cost of Sales',Cost_GL_ACno),

price_level_accounting_switch(PLA_type),

( PLA_type = 'Historical Cost',

/* indexed_assert jel P_and_I_item(JEno,Cost_GL_ACno,Money_value,Date), */

/* Debit CoGS */

assert_uncondit_jel(JEno,Cost_GL_ACno,UFno1),

/* Credit Inventory */

Credit_amount is -Money_value,

assert_jel_via_AO_mapping(JEno,AO_ACno,Credit_amount,Date_acq)

/* Debit Cost of Sales: a(sl/s0)(g/g0) */

assert_uncondit_jel(JEno,Cost_GL_ACno,UFno1),

get_price_index(g,Date_acq,Gen_index_acq),

Coef is Money_value/Gen_index_acq,

assertz(uncondit_formula(UFno1,Coef,[[g,1]])),

/* Credit Inventory -a(g/g0) */

Credit_amount is -Money_value,

assert_jel_via_AO_mapping(JEno,AO_ACno,Credit_amount,Date_acq)

/* Debit Cost of Sales: a(sl/s0)(g/g0) */

assert_uncondit_jel(JEno,Cost_GL_ACno,UFno1),

CoGS_coeff is Money_value*(Spec_index_sale/Spec_index_acq)/

Gen_index_sale,

assertz(uncondit_formula(UFno1,CoGS_coeff,[[g,1]])),

/* Credit Inventory: a(s/s0) */

map_AOac_to_GLac(AO_ACno,Invent_GL_ACno),

Invent_coeff is -Money_value/Spec_index_acq,

assertz(uncondit_formula(UFno1,Invent_coeff,[[Series,1]])),

/* now assert je to lock in gain above general prices */

/* (a(g1/g0) - a(s/s0)) */

map_AC_concept_to_GLac('Real Holding Gain',RHG_GL_ACno),

HG_coeff_S is -Invent_coeff,

HG_coeff_G is -CoGS_coeff,

Event_handler - 4
assert_uncondit_jel(JEno,RHG_GL_ACno,_,UFnoHG),
assertz(uncondit_formula(UFnoHG,HG_coeff_G,[[g,1]])),
assertz(uncondit_formula(UFnoHG,HG_coeff_S,[[Series,1]]))

write('PLA_type is '),write(PLA_type),nl,
write('Error: Credit sale not programmed for other than hc., cpp., ')
write('or cca. accounting. Abort and correct.'),nl

assert_jels_for_inventory(JEno,AO_ACno,Date,T,Short).

/* TEST DATA (not used at present)*/
eh(1,870801,'Credit Purchase','Supplier Invoice',785,'null','Purchase 3 AL350 bench lathes').
rc(1,1,11401,3_,_).
rc(1,2,21101,-300_,_).

eh(2,870802,'Credit Purchase','Supplier Invoice',785,'null','Purchase 3 AL350 bench lathes').
rc(2,1,11401,3_,_).
rc(2,2,21101,-3033_,_).

eh(3,870803,'Credit Purchase','Supplier Invoice',785,'null','Purchase 3 AL350 bench lathes').
rc(3,1,11401,3_,_).
rc(3,2,21101,306_,_).

eh(4,870804,'Credit Sale','Sales Invoice',233,'null','Sale to Hardware Manufacturing').
rc(4,1,11204,10003_,_).
rc(4,2,11401,-2_,_).

eh(5,870804,'Credit Sale','Sales Invoice',233,'null','Sale to Hardware Manufacturing').
rc(5,1,11204,10003_,_).
rc(5,2,11401,-2_,_).

eh(6,870804,'Credit Sale','Sales Invoice',233,'null','Sale to Hardware Manufacturing').
rc(6,1,11204,10003_,_).
rc(6,2,11401,-4_,_).

*/

/* CODE BELOW NOT USED */
npv(Rno,Rtype,Dollar_value,Money_value):-
  ( is_a(resource_hierarchy,Rtype,'Trade Debt'),
    Money_value is Dollar_value
  ; write('not yet coded for discounting cash flows'),nl
  )

/* DONATION */
donation(Eno,Date,Etype,Docno,AgentID,Narrative,JEno):-
  write('donation called for event '),write(Eno),nl,
  assert_jeh(JEno,Date,Eno,Narrative),
  loop_through_event_detail_lines(JEno,Eno,Date), /* always true */
  write('The balancing item for all jes so far must be donation expense */
  check_je_balances(JEno,'Balance formulae required'),

Event_handler - 5
once((fast_setof(Coeff,balance_formula_regd(_,_,Coeff,_,J,Coeffs),
    write('Coeffs for balance formula ='),write(Coeffs),nl,*/
    (Coeffs==[])

  map_AC_concept_to_GLac('Donation Expense',GL_ACno),
  assert_balancingje_lines(JEno,GL_ACno)
))

Once((fast_setof(Coeff,balance_formula_regd(_,_,Coeff,_,J,Coeffs),
    write('Coeffs for balance formula ='),write(Coeffs),nl,*/
    (Coeffs==[])

  map_AC_concept_to_GLac('Donation Expense',GL_ACno),
  assert_balancingje_lines(JEno,GL_ACno)
))

The code below follows case One.

price_level_accounting_switch(PLA_type),
  (PLA_type = 'Historical Cost',
    Money_value is Par_value * Q, /* i.e., uninflated Par value */
    map_AOac_to_GLac(AO_ACno,GL_ACno),
    assert_uncondit_jel(JEno,GL_ACno,Fno,UFno),
    assertz(uncondit_formula(UFno,Money_value([]))),
    /* calculate share premium or discount today */
    Share_prem_disc_per_share is V/Q - Par_value,
    write("Share_prem_disc_per_share is "),write(Share_prem_disc_per_share),nl,*/
    (Share_prem_disc_per_share == 0, true /* do nothing*/)
Event_handler - 6
252  Share_prem_disc_per_share @> 0,
253  map_AC_concept_to_GLac('Share Premium',SPD_GL_ACno),
254  Sh_prem_disc_money value is Share_prem_disc_per_share * Q,
255  assert_unconditional(JEno,SPD_GL_ACno,SPD_Fno,SPD_UFno),
256  assertz(unconditional_formula(SPD_UFno,Sh_prem_disc_money_value,[[]]))
257
258  map_AC_concept_to_GLac('Share Discount',SPD_GL_ACno),
259  Sh_prem_disc_money value is Share_prem_disc_per_share * Q,
260  write('Share Discount is '),write(Sh_disc_money_value),nl,
261  assert_unconditional(JEno,SPD_GL_ACno,SPD_Fno,SPD_UFno),
262  assertz(unconditional_formula(SPD_UFno,Sh_disc_money_value,[[]]))
263
264  (PLA_type = 'Current Purchasing Power';
265    PLA_type = 'Current Cost Accounting'),
266  map_AOAC_to_GLac(AO_ACno,GL_ACno),
267  get_price_index(g,Date_incorporated,Gen_index_value_at_incorp),
268  Coeff is Par_value*Q/Gen_index_value_at_incorp,
269  assertz(unconditional_formula(UFno,Coeff,[[g,1]])),
270  /* calculate share premium or discount today using indexed Par val*/
271  get_price_index(g,Date,Gen_index_value),
272  Share_prem_disc_per_share is V/Q -
273  Par_value*Gen_index_value/Gen_index_value_at_incorp,
274  ( Share_prem_disc_per_share == 0,
275    true /* do nothing*/
276  ;
277  Share_prem_disc_per_share @> 0,
278  map_AC_concept_to_GLac('Share Premium',SPD_GL_ACno),
279  assert_unconditional(JEno,SPD_GL_ACno,SPD_Fno,SPD_UFno),
280  SPD_Coeff is Share_prem_disc_per_share*Q/Gen_index_value,
281  assertz(unconditional_formula(SPD_UFno,SPD_Coeff,[[g,1]])),
282
283  map_AC_concept_to_GLac('Share Discount',SPD_GL_ACno),
284  assert_unconditional(JEno,SPD_GL_ACno,SPD_Fno,SPD_UFno),
285  SPD_Coeff is Share_prem_disc_per_share*Q/Gen_index_value,
286  assertz(unconditional_formula(SPD_UFno,SPD_Coeff,[[g,1]])),
287  Sh_disc_money_value is Share_prem_disc_per_share * Q,
288  write('Share Discount is '),write(Sh_disc_money_value),nl,
289  true
290  )
291
292  ;
293  write('PLA_type is '),write(PLA_type),nl,
294  write('Error: not programmed for other than hc., cpp., or '),
295  write('cca. accounting. Abort and correct.'),nl
296
297  /*rely on processing seq for ELSE */
298  (assert_core Je_line(Eno,Lno,JEno,Date,AO_ACno,Q,V);true)
299  ),
300  fail.
301  loop_through_issue_shares_event_detail_lines(JEno,Eno,Date). /* always true */
302  /* -- left over from 1986 --
303  assert_share_premium_or_discount(JEno);
304
305  Event_handler - 7
sum(Amount, jel(JEno,_,_,Amount), Journal_entry_balance),

(Journal_entry_balance > 0,
 Share_premium_reserve is -Journal_entry_balance,
 write('Share Premium Reserve is '),write(Share_premium_reserve),nl,
 map_AC_concept_to_GLac('Share Premium Reserve', GL_ACno),
 assert_prior_or_current_period_jel(JEno, Date, GL_ACno, Share_premium_reserve)
 ;

(Journal_entry_balance < 0,
 Share_discount is -Journal_entry_balance,
 write('Share Discount is '),write(Share_discount),nl,
 map_AC_concept_to_GLac('Share Discount', GL_ACno),
 assert_prior_or_current_period_jel(JEno, Date, GL_ACno, Share_discount)
 );

/*--------- MANUFACTURING EVENTS -----------------------------*/

manuf_event(Eno, Date, Etype, Doctype, Docno, AgentID, Narrative, JEno):-

write('manuf event called for event '),write(Eno),nl,
assert_jel(JEno, Date, Eno, Narrative),
loop_through_manuf_event_detail_lines(JEno, Eno, Date).

/* always true */

loop_through_manuf_event_detail_lines(JEno, Eno, Date):-
ed(Eno, Lno, AO_ACno, Q, V),
    assert_manuf_je_line(Eno, Lno, JEno, Date, AO_ACno, Q, V).

/* always true */

loop_through_manuf_event_detail_lines(JEno, Eno, Date). /* always true */

assert_manuf_je_line(Eno, Lno, JEno, Date, AO_ACno, Q, V):-

once(determine_money_value_for_core_je_line(Eno, JEno, Date, AO_ACno, Q, V, Money_value, Money_date)),
    /* (the above procedure isolates any std. cost price variances) */
    once(assert_jel_via_production_step(JEno, Date, AO_ACno, Money_value, Money_date)),
    /* remaining code from assert_core_je_line, i.e., for depreciation and inventory, deleted */
    fail.

assert_jel_via_production_step(JEno, Date, AO_ACno, Money_value, Money_date):-
    /* write(assert_jel_via_production_step(JEno, Date, AO_ACno, Money_value, Money_date)),nl, */
    /* first check for normal AO_mapping; if not, then use production step */
    assert_jel_via_AO_mapping(JEno, AO_ACno, Money_value, Money_date),
    /* for manufacturing, map_AOac_to_GLac replaced by production_step tables */
    account(AO_ACno, Rno, Rtype),
    resource(Rno, Rtype),
    once(
        production_step_input_AOac(AO_ACno, Job_no, Step_no),
        production_step_GLac(Job_no, Step_no, M/GLac, L/GLac, OH_Incurred/GLac, OH_Applied/GLac),
        (is_a(resource_hierarchy, Rtype, 'Tangible Resource'),
         /* materials into a production step includes M+L+OH product from */
         /* a previous production step */
         indexed_assert_jel_Rtype(JEno, M/GLac, Money_value, Money_date, Rtype))
    );
is_a(resource_hierarchy,Rtype,'Labour Services'),
indexed_assert_jel_Rtype(JEno,L_GLac,Money_value,Money_date,Rtype)
;
indexed_assert_jel_Rtype(JEno,OH_Incurred_GLac,Money_value,Money_date,Rtype)
)
)
production_step_output_AOac(AO_ACno,Job_no,Step_no),
production_step_GLacs(Job_no,Step_no,M_GLac,L_GLac,OH_Incurred_GLac,
  OH_Applied_GLac),
/* get material, labour, and overhead cost components */
max(D_set,
  ( derived_unit_standard_cost(Rno,D_set,_,_,_,,,$
    Date @$= D_set$
  ),
    Date_set),
  derived_unit_standard_cost(Rno,Date_set,Std_cost,M_std_cost,
    L_std_cost,OH_std_cost),
  M_value is Money_value * M_std_cost / Std_cost,
  L_value is Money_value * L_std_cost / Std_cost,
  OH_Applied_value is Money_value * OH_std_cost / Std_cost,
  /* Can't easily generate a weighted index for M, L, and OH, so
  index these component costs with the Rtype of the product!!! */
  /* that way the transfer to next dept or Finished Goods je balances*/
  /* differences will be caught as Inter-index gain in Production events*/
indexed_assert_jel_Rtype(JEno,M_GLac,M_value,Money_date,Rtype),
indexed_assert_jel_Rtype(JEno,L_GLac,L_value,Money_date,Rtype),
indexed_assert_jel_Rtype(JEno,OH_Applied_GLac,OH_Applied_value,
  Money_date,Rtype)
)
)
/*---------- OPEN ACCOUNTS ----------------------------------------*/
open_acs(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno):-
  write('open_acs called for event '),write(Eno),nl,
  assert_jeh(JEno,Date,Eno,Narrative),
  loop_through_open_acs_event_detail_lines(JEno,Eno,Date),/* always true */
  assert_provision_for_doubtful_debts(JEno,Date), /* Report date money values*/
  assert_opening_balances_fudge(JEno).

loop_through_open_acs_event_detail_lines(JEno,Eno,Date):-
ed(Eno,Lno,AO_ACno,Q,V), /*step through the event detail lines */
  /* isolate own share-holders funds for special treatment: focus */
  /* on the liability on liquidation, which is monetary in nature */
  once(
    ( ao_account(AO_ACno,,'Owners' Equity',_,_),
      map_AOac_to_GLac(AO_ACno,GL_ACno),
      indexed_assert_jel_P_and_L_item(JEno,GL_ACno,V,Date)
    /* no gearing gain on own shareholders */
  ), /*rely on processing seq for ELSE */
  (assert_core_je_line(Eno,Lno,JEno,Date,AO_ACno,Q,V);true)
  ),
  fail.
loop_through_open_acs_event_detail_lines(JEno,Eno,Date). /* always true */
assert opening_balances_fudge(JEno):-
/* The balancing item for all jes so far is Retained earnings, etc.*/
check_je_balances(JEno,'Balance formulae required'),
once((fast_setof(Coeff,balance_formula_regd(_,_,_,_,_,_,Coeffs)),
/* write('Coeffs for balance formula ='),write(Coeffs),nl, */
(Coeffs=[]=0)
;)
map_AC_concept_to_GLac('Opening Balance Fudge',Fudge_GL_ACno),
assert_balancing_je_lines(JEno,GL_ACno)
).

/******** PRODUCTION SUMMARY ************/
production_summary(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno):-
write('production_summary called for event '),write(Eno),nl,
assert_jeh(JEno,Date,Eno,Narrative),
/* Could check here to see that all AO Accounts in production steps*/
/* sum to zero. For time being, simple close material and labour to P&L*/
fast_setof((Job_no,Step_no),
( ed(Eno,Lno,AO_ACno,Q,V),
 production_step_input_AOac(AO_ACno,Job_no,Step_no);
 production_step_output_AOac(AO_ACno,Job_no,Step_no)));
List_of_Job_Steps),
write(List_of_Job_Steps),nl,
close_production_step_GLacs(JEno,List_of_Job_Steps).
close_production_step_GLacs(JEno,[]).
close_production_step_GLacs(JEno,[(Job_no,Step_no)|Remaining_list_of_Job_Steps]),
/* write(close_production_step_GLacs(JEno,[(Job_no,Step_no)|Remaining_list_of_Job_Steps])),nl, */
production_step_GLacs(Job_no,Step_no,M_GLac,L_GLac,OH_Incurred_GLac,
OH_Applied_GLac),
map_AC_concept_to_GLac('Profit and Loss',PL_GLac),
assert Jets to close account(M_GLac,PL_GLac,JEno),
assert Jets to close account(L_GLac,PL_GLac,JEno),
/* Can't close the overhead accounts since they exist for each Department*/
/* not for each job step, e.g., can't link depreciation of machine to job*/
/* since m/c may be used for many jobs in one month. Therefore, can only*/
/* calculate OH variances at the end of each period. */
close_production_step_GLacs(JEno,Remaining_list_of_Job_Steps).

/******** SALE OF DEPRECIABLE ASSET *************/
sale_depreciable_asset(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno).
sale_depreciable_asset(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno):-
write('sale_depreciable_asset called for event '),write(Eno),nl,
assert_jeh(JEno,Date,Eno,Narrative),
loop_through_sale_da_event_detail_lines(JEno,Eno,Date), /*always true */
/* now treat balancing amount as profit or loss on disposal */
map_AC_concept_to_GLac('Non-operating Income',GL_ACno),
assert_balancing_je_lines(JEno,GL_ACno).
loop_through_sale_da_event_detail_lines(JEno,Eno,Date):-
Event_handler - 10
ed(Eno,Lno,AO_ACno,Q,V), /*step through the event detail lines */
write(ed(Eno,Lno,AO_ACno,Q,V)),nl, /*
once(ac_account(AO_ACno,,Rno,,)),
once(resource(Rno,,Rtype)),
map_AOac_to_GLac(AO_ACno, GL_ACno), /* first transfer "balance" of Amort/Depr Expense to P&L */
map_AC_concept_to_GLac('Profit and Loss',PL_ACno),
once(

Q @< 0,
is_a(resource_ev_hierarchy,Rtype,'Shrinking Value Resource'),
/* is_a(resource_ev_hierarchy,Rtype,'Tangible Resource'),
write('about to start jels for disposal of depr asset'),nl, */
gl_depreciation(GL_ACno,Cr_ACno,Dr_ACno),
assert_close depre expense_jel(JEno,Date,Dr_ACno,PL_ACno), /* first transfer present depre exp balance to Profit and Loss*/
now remove FORMULAE Cr Depr Expense, Dr Accum Depr, Cr Asset */
assert_jels_to_close_account(Dr_ACno,_,JEno), /* Cr Depr Exp*/
/* above procedure is in file: formula */
assert_jels_to_close_account(GL_ACno,_,JEno), /* Cr Asset */
assert_depr_ac_for_asset(GL_ACno,Cr_ACno),/*
assert_jels_to_close_account(Cr_ACno,_,JEno) /* Dr Accum Depr */
/* assert_balancing_jels will treat the rest as non-op P or L */
assert_balancing_jels will treat the rest as non-op P or L */
is_a(resource_ev_hierarchy,Rtype,'Intangible Resource'),
write('Sale of amortizable asset, does this make sense?'),nl,
gl_amortization(GL_ACno,Dr_ACno),
/* first clear the balance, then remove formula for amort exp */
/* assert_balancing_jels will do the rest*/
assert_close depre expense_jel(JEno,Date,Dr_ACno,PL_ACno),
assert_jels_to_close_account(Dr_ACno,_,JEno) /* Cr Amort Exp*/
assert_core_jel_line(Eno,Lno,JEno,Date,AO_ACno,Q,V)
assert_core_jel_line(Eno,Lno,JEno,Date,AO_ACno,Q,V)
assert_close depre expense_jel(JEno,Date,GL_ACno,PL_ACno)-
assert_close depre expense_jel(JEno,Date,PL_ACno)-
/* may be more than one */
indexed_assert_jel_P_and_L_item(JEno,PL_ACno,F_value,Date)
assert_balancing_jels will do the rest*/
assert_close depre expense_jel(JEno,Date,GL_ACno,PL_ACno).
assert_close depre expense_jel(JEno,Date,GL_ACno,PL_ACno).

END OF EVENT_HANDLER

done(ehandler).
<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>assert_closing_depr_expense jealousy(JEno,Date,GL_ACno,PL_ACno)</td>
<td>11</td>
</tr>
<tr>
<td>assert_credit_sale je line(JEno,AO_ACno,Q,V,Date)</td>
<td>3</td>
</tr>
<tr>
<td>assert_jel_via_production_step(JEno,Date,AO_ACno,Money_value,Money_value)</td>
<td>8</td>
</tr>
<tr>
<td>assert_jeis_for_inventory(JEno,AO_ACno,Date,</td>
<td>H</td>
</tr>
<tr>
<td>assert_manuf je line(Eno,Lno,JEno,Date,AO_ACno,Q,V)</td>
<td>8</td>
</tr>
<tr>
<td>assert_opening_balances_fudge(JEno)</td>
<td>10</td>
</tr>
<tr>
<td>assert_share_premium_or_discount(JEno)</td>
<td>7</td>
</tr>
<tr>
<td>cash_payment(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno)</td>
<td>2</td>
</tr>
<tr>
<td>cash_receipt(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno)</td>
<td>2</td>
</tr>
<tr>
<td>close_production_step_GLac(JEno,[(Job_no,Step_no)] Remaining_inventory)</td>
<td>10</td>
</tr>
<tr>
<td>credit_sale(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno)</td>
<td>2</td>
</tr>
<tr>
<td>donation(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno)</td>
<td>5</td>
</tr>
<tr>
<td>issue_shares(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno)</td>
<td>6</td>
</tr>
<tr>
<td>loop_through_issue_shares_event_detail_lines(JEno,Eno,Date)</td>
<td>6</td>
</tr>
<tr>
<td>loop_through_manuf_event_detail_lines(JEno,Eno,Date)</td>
<td>8</td>
</tr>
<tr>
<td>loop_through_open_acs_event_detail_lines(JEno,Eno,Date)</td>
<td>9</td>
</tr>
<tr>
<td>loop_through_sale_depreciable_event_detail_lines(JEno,Eno,Date)</td>
<td>10</td>
</tr>
<tr>
<td>manuf_event(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno)</td>
<td>8</td>
</tr>
<tr>
<td>npv(Rno,Rtype,Dollar_value,Money_value)</td>
<td>5</td>
</tr>
<tr>
<td>open_acs(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative,JEno)</td>
<td>9</td>
</tr>
<tr>
<td>production_summary(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative)</td>
<td>10</td>
</tr>
<tr>
<td>sale_depreciable_asset(Eno,Date,Etype,Doctype,Docno,AgentID,Narrative)</td>
<td>10</td>
</tr>
<tr>
<td>sda</td>
<td>10</td>
</tr>
</tbody>
</table>
Index to Procedures in Event_handler

assert_closing_depr_expense_line(JEno,Date,GL_ACno,PL_ACno) .................. 11
assert_credit_sale_line(JEno,OA_ACno,Q,V,Date) .......................... 3
assert_jel_via_production_step(JEno,Date,AO_ACno,M,Money_value) .......... 8
assert_jels_for_inventory(JEno,OA_ACno,Date,[H,T],Shortfall) ......... 3
assert_manuf_jel_line(Eno,Eno,JEno,Date,AO_ACno,Q,V) ................. 8
assert_opening_balances_fudge(JEno) ................................... 10
assert_share_premium_or_discount(JEno) .................................. 7
cash_payment(Eno,Date,Doctype,Docno,AgentID,Narrative,JE) .......... 2
cash_receipt(Eno,Date,Doctype,Docno,AgentID,Narrative,JE) .......... 2
close_production_step_GLacs(Eno,(Job no,Step no),Remaining) ........ 10
credit_sale(Eno,Date,Doctype,Docno,AgentID,Narrative,JE) .......... 2
donation(Eno,Date,Doctype,Docno,AgentID,Narrative,JE) .......... 5
issue_shares(Eno,Date,Doctype,Docno,AgentID,Narrative,JE) .......... 6
loop_through_issue_shares_event_detail_lines(JEno,Eno,Date) .......... 6
loop_through_manuf_event_detail_lines(JEno,Eno,Date) ................. 8
loop_through_open_acs_event_detail_lines(JEno,Eno,Date) ............. 9
loop_through_sale_da_event_detail_lines(JEno,Eno,Date) ............... 10
manuf_event(Eno,Date,Doctype,Docno,AgentID,Narrative,JE) .......... 8
npv(Rno,Rtype,Dollar_value,Money_value) .................................. 5
open_acs(Eno,Date,Doctype,Docno,AgentID,Narrative,JE) .............. 9
production_summary(Eno,Date,Doctype,Docno,AgentID,Narrative) .... 10
sale_depreciable_asset(Eno,Date,Doctype,Docno,AgentID,Narrative) . 10
sda ......................................................... 10

Event_handler - 12
APPENDIX 4: PROLOG LIBRARY

This Appendix contains 350 lines (8 pages) of Prolog "Library" code. Procedures in this library are used by the higher-level procedures in the earlier Appendices. They are, in effect, extensions to the basic procedures defined by Clocksin and Mellish [1981]. Much used procedures are:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>once(X)</td>
<td>2</td>
</tr>
<tr>
<td>retractall(X)</td>
<td>2</td>
</tr>
<tr>
<td>sum(X,G,S)</td>
<td>3</td>
</tr>
<tr>
<td>fast_setof(Template,Goal,Set)</td>
<td>3</td>
</tr>
<tr>
<td>round_X_to_two_dec_places(X,Rounded_X)</td>
<td>4</td>
</tr>
<tr>
<td>write_fixed_length(Field_name,Length)</td>
<td>5</td>
</tr>
<tr>
<td>is_a(Hierarchy,Descendant,Ancestor)</td>
<td>5</td>
</tr>
<tr>
<td>date_function(D,Y,F,Target_date,Function_value)</td>
<td>6</td>
</tr>
<tr>
<td>number_of_days_from_to(Date_from,Date_to,N)</td>
<td>8</td>
</tr>
</tbody>
</table>

There is an index to all procedures in this Appendix on page 10.
once(X) :- call(X), !.

sum([],S) :- S is 0.
sum([H|T],S) :- sum(T,S1), S is H + S1.

cardinality([],C) :- C is 0. /* C is the number of members in a list */
cardinality([H|T],C) :- cardinality(T,C1), C is 1 + C1.

/* retractall, p. 179 Clocksin and Mellish */
retractall(X) :-
    retract(X), fail.
retractall(X) :-
    retract((X:-Y)), fail.
retractall(X).

/* append, page 63 Clocksin and Mellish */
append([],L,L).
append([X|L1],L2,[X|L3]) :- append(L1,L2,L3).

/* set manipulations from pp153,4 Clocksin and Mellish */
member(X,[X|_]).
member([X|Y],_):-member(X,Y).
subset([A|X],Y):-member(A,Y),subset(X,Y).
subset([],Y).

intersection([],X,[]).
intersection([X|R],Y,[X|Z]) :- member(X,Y),!
intersection(R,Y,Z).

union([],X,X).
union([X|R],Y,Z) :- member(X,Y),!
union(R,Y,Z).

/* set_diff(A,B,C):- ;; any members of A not in B are listed in C ... pbs */
set_diff([A|X],B,C) :- member(A,B),!
set_diff(X,B,C).
set_diff([A|X],B,[A|C]) :- set_diff(X,B,C).
set_diff([],B,[]).

/* AGGREGATE FUNCTIONS */

/* format: agg(X,G,R), where G is a goal of arbitrary complexity */
/* (returns mark if G does not exist). */

e.g., max(Eno,inventory_value(_,Eno,_,_),Eno-max) finds max Eno in
a FIFO table of inventory values.

Prolog_library - 2
The first line repeatedly calls G and stores the variable specified as X in agg(X,G,R) in found(...). Using the example above, if G is inventory_value(Rno,Eno,Y,_,_), then it is Eno's, not Rno's or Y's that are stored as found(Eno), because X has value Eno.

cnt(X,G,:) :- asserta(found(mark)), call(G), asserta(found(X)), fail.
cnt(X,G,C) :- cnt_found(0,C),!.

sum(X,G,:) :- write_down(X,G), fail.
sum(X,G,S) :- sum_found(0,S),!.

avg(X,G,:) :- write_down(X,G), fail.
avg(X,G,A) :- avg_found(0,A),!.

max(X,G,:) :- write_down(X,G), fail.
max(X,G,M) :- found(Top), max_found(Top,M),!.

min(X,G,:) :- write_down(X,G), fail.
min(X,G,M) :- found(Top), min_found(Top,M),!.

cnt_found(I,J) :- getnext(X), K is I+1, cnt_found(K,J).
cnt_found(K,K).

sum_found(I,J) :- getnext(X), K is I+X, sum_found(K,J).
sum_found(K,K).

avg_found(I,T,K) :- getnext(X), M is I+1, N is J+X, avg_found(M,N,K).
avg_found(M,N,A) :- A is N/M,i.

max_found(I,J) :- getnext(X), (X> I, max_found(X,J); max_found(I,J)).
max_found(K,K).

min_found(I,J) :- getnext(X), (X<I, min_found(X,J); min_found(I,J)).
min_found(K,K).

sum_found(I,J) :- getnext(X), K is I+X, sum_found(K,J).
sum_found(K,K).

write_down(X,G) :- asserta(found(mark)), call(G), asserta(found(X)).

getnext(X) :- retract(found(X)), l, X \= = mark.

/* pbs coded FAST_SETOF -------------------------------------------*/
/* Prolog's setof predicate creates a set from a set of asserted facts. */
/* Fast_setof calls the Goal repeatedly, and then asserts the relevant */
/* parts of the Goal found. Prolog's setof is then used to harvest the set */

fast_setof(Template,Goal,Set):-
  write_down(Template,Goal),fail.
  /* uses asserta to assert found(mark), found(1st instance), found(2nd.. etc.*/) 

fast_setof(Template,Goal,Set):-
  listing(found),nl, *
  get_fast_set(Set),abolish(fast_set,1),!.

Prolog_library - 3
/* retracts "found's" down to found(mark) and asserts fast_set(nth instance).*/
/* then uses setof to collect, and finally abolish, the fast_set()'s*/

get_fast_set(Set):-
  getnext(Template),asserta(fast_set(Template)),get_fast_set(Set).
get_fast_set(Set):-
  setof(Template,fast_set(Template),Set).

/*-------------------------- ROUND TO n DECIMAL PLACES --------------------------*/
/* converts to cents, as well, to avoid 15 signif. digits on CYBER */
r(X):-round(X to two dec_places(X,Rounded_X),write(Rounded_X),nl.
/r(X,N):-round(X to N dec_places(X,N,Rounded_X),write(Rounded_X),nl.*

round_X_to_two_dec_places(X,Rounded_X):-
  /
  tell('number from_prolog'),write(X),told,
  system("exet obj round_to_2_dec_places").
  see('return_to_prolog'),read(Rounded_X),seen.
  /
  ( X > -0.0000000001,
    X < 0.0000000001,
    Rounded_X is 0
  ;
    Hundred_X is X * 100,
    Floor_hundred_X is floor(Hundred_X),
    Rem is Hundred_X - Floor_hundred_X,
    ( Rem > 0.5,
      Rounded_X is Floor_hundred_X + 1
    ;
      Rounded_X is Floor_hundred_X
  ).
  ).
/*
100 open "number_from_prolog" for input as #1
102 open "return_to_prolog" for output as #2
103 300 input #1, N
104 400 print #2, using "############.##_; N
106 600 end
109 */

round_X_to_two_dec_places(X,Rounded_X):-
  round_X_to_N_dec_places(X,2,Rounded_X).
round_X_to_N_dec_places(X,N,Rounded_X):-
  
  round_positive_X_to_N_decimal_places(X,N,Rounded_X).
  
  X <= 0,
  round_positive_X_to_N_decimal_places(X,N,Rounded_X)
  ;
  X < 0,
  Y is - X,
  round_positive_X_to_N_decimal_places(Y,N,Rounded_Y),
  Rounded_X is - Rounded_Y

Prolog_library - 4
round_positive(X,10^N, Rounded_X).
Super_X is X*10^N,
Trunc_Super_X is floor(Super_X),
Rem is Super_X - Trunc_Super_X,
(Rem < 0.5,
    Rounded_Super_X is Trunc_Super_X)
Rem >= 0.5,
    Rounded_Super_X is Trunc_Super_X + 1
)
Rounded_X is Rounded_Super_X/(10^N).

/* WRITE FIXED LENGTH */
write_fixed_length(Field_name,Length):-
/\ truncates, or adds spaces to pad out the field, to Length chars/\nname(Field_name,Ascii_list),
put_list(Ascii_list,Var_len,Length,0), /* returns Var_len*/
Number_of_spaces is Length - Var_len,
(Number_of_spaces > 0,
    put_spaces(Number_of_spaces)
); true.

put_list([],0,_,_). /* sets Var_len to 0 before unwinding*/
put_list(_0,Length,Length). /* truncates remaining chars, then unwind*/
put_list([H|Remainder_list],L,Length,Chars_done):-
put(H),
Chars_done_plus_1 is Chars done + 1,
put_list(Remainder_list,L_minus_1,Length,Chars_done_plus_1),
L is L_minus_1 + 1.

put_spaces(Number_of_spaces):-
write(" "),
One_less is (Number_of_spaces - 1),
( One_less > 0,
    put_spaces(One_less)
); true.

/* IS_A */
is_a(Hierarchy,Descendant,Descendant). /* for when the node is a hole in one*/
is_a(Hierarchy,Descendant,Ancestor):-
once(Hierarchy(_Parent,Descendant)),
( Parent == Ancestor

Prolog_library - 5
is_a(Hierarchy, Parent, Ancestor).

is_a(Hierarchy, Descendant, Ancestor):-
ancestor_list(Hierarchy, Descendant, L),
\write(L),
!,
member(Ancestor, L).

ancestor_list(Hierarchy, Descendant_node, L):-
Hierarchy(_, Parent_node, _, Descendant_node),
ancestor_list(Hierarchy, Parent_node, SubL),
append([Descendant_node], SubL, L).
ancestor_list(Hierarchy, Root, [Root]).

\/*
Y is a contin. real function of D. F is a predicate for time series for Y */
date_function(D,Y,F,Target_date,Function_value):-
\/* test to see if function is already defined for target date */
D = Target_date, 
call(F),
Function_value = Y. /* NB. not == */
\/* write('direct hit'),nl. */

\/*
check if function value has already been saved, if not, go the long way: */
\( saved_function(D,Y,F,Target_date,Function_value);true),
\( nonvar(Function_value),
\( write('using saved function value'),nl, */
true
\);
\( true
\)
\( var(Function_value),
date_function_long_way(D,Y,F,Target_date,Function_value),
assert(saved_function(D,Y,F,Target_date,Function_value))
\).

date_function_long_way(D,Y,F,Target_date,Function_value):-
\( retractall(previous_series(_,_)),
\( retractall(future_series(_,_))
\),
\/* split the time series into the "befores" and the "afters" */
call(F),
\( once(( Target_date > D,
\( asserta(previous_series(D,Y))
\)),
\( Target_date < D,
\( assertz(future_series(D,Y))
\) /* Note: tested for the "==" case above.... direct hit */
\)),
\fail.

date_function_long_way(D,Y,F,Target_date,Function_value):-

Prolog_library - 6
/* listing(previous_series), */
/* listing(future_series), */
(retract(previous_series(Previous_date,Previous_value));true),
(retract(future_series(Next_date,Next_value));true),
/* write(previous_series(Previous_date,Previous_value)),nl */
/* write(future_series(Next_date,Next_value)),nl */
(var(Previous_date),var(Next_date),
write('No predicate exists for time series: '),write(F),nl)

(var(Previous_date),nonvar(Next_date),
(retract(future_series(Next_next_date,Next_next_value));true),
(var(Next_next_date),
write('Only one entry in time series for: '),write(F),
write('. Function set to this value.'),nl,
Function_value is Next_value)

(nonvar(Next_next_date),
number_of_days_from_to(Next_date,Next_next_date,Days_to_next),
Function_change_per_day is (Next_next_value - Next_value)/Days_to_next,
number_of_days_from_to(Target_date,Next_next_date,Days_from_target),
write('Extrapolating back '),write(Days_from_target),
write(' days for time series. (Suspect for long times.)'),nl,
Function_value is Next_value - Function_change_per_day * Days_from_target)

(nonvar(Previous_date),var(Next_date),
(retract(previous_series(Previous_previous_date,Previous_previous_value));true),
(var(Previous_previous_date),
write('Only one entry in time series for: '),write(F),
write('. Function set to this value.'),nl,
Function_value is Previous_value)

(nonvar(Previous_previous_date),
number_of_days_from_to(Previous_previous_date,Previous_date,Days_apart),
number_of_days_from_to(Previous_date,Target_date,Days_to_target),
write('Extrapolating forward '),write(Days_to_target),
write(' days for time series. (Suspect for long times.)'),nl,
/* No, the 100 4' below is not nonsense; it is a way round a suspected BUG in Prolog interpreter */
Function_val is 100 * Previous_value + 
100 * (Previous_value - Previous_previous_value) / Days_apart * Days_to_target,
Function_value is Function_val / 100)

(nonvar(Previous_date),nonvar(Next_date),
/* interpolate for function value */
number_of_days_from_to(Previous_date,Next_date,Days_to_next),
number_of_days_from_to(Previous_date,Target_date,Days_to_target),
Function_val is 100 * Previous_value + 
100 * (Next_value - Previous_value) / Days_to_next * Days_to_target,
Function_value is Function_val / 100).

/*---------------- NUMBER OF DAYS, MONTHS, YEARS --------------------------*/
day(Date,Da):- Da is Date mod 100.
282  month(Date,Mo):- YrMo is Date // 100, /* truncates the dec. fraction*/
283     Mo is YrMo mod 100.
284  
285  year(Date,Yr):- Yr is Date // 10000.
286
287  number_of_days_from_to(Date_from,Date_to,N):-
288     ( Date_from > Date_to,
289      write('Start date:'), write(Date_from), write(' less than end date: '),
290      write(Date_to), nl,
291      number_of_days_from_to(Date_to,Date_from,Pos_N),
292      N is -Pos_N
293     ;
294     Date_from == Date_to,
295     N is 0
296     ;
297     year(Date_to,Yr_to),
298     year(Date_from,Yr_from),
299     month(Date_to,Mo_to),
300     month(Date_from,Mo_from),
301     day(Date_to,Da_to),
302     day(Date_from,Da_from),
303     days_to_start_month(Mo_to,Days_to_start_month_to),
304     days_to_start_month(Mo_from,Days_to_start_month_from),
305     Real_N is (Yr_to - Yr_from) * 365.25
306            + (Days_to_start_month_to - Days_to_start_month_from)
307            + (Da_to - Da_from),
308      N is floor(Real_N)
309  ).
310  
311  days_to_start_month(1,0).
312  days_to_start_month(2,31).
313  days_to_start_month(3,59).
314  days_to_start_month(4,90).
315  days_to_start_month(5,120).
316  days_to_start_month(6,151).
317  days_to_start_month(7,181).
318  days_to_start_month(8,212).
319  days_to_start_month(9,243).
320  days_to_start_month(10,273).
321  days_to_start_month(11,304).
322  days_to_start_month(12,334).

323  number_of_months_from_to(Date_from,Date_to,N):-
324     year(Date_to,Yr_to),
325     year(Date_from,Yr_from),
326     month(Date_to,Mo_to),
327     month(Date_from,Mo_from),
328     day(Date_to,Da_to),
329     day(Date_from,Da_from),
330     ( Da_to > Da_from,
331        N is (Yr_to - Yr_from) * 12 + (Mo_to - Mo_from)
332        ;
333        Da_to < Da_from,
334        N is (Yr_to - Yr_from) * 12 + (Mo_to - Mo_from) - 1
335    ),
336    !,
337    (!N<0,write('Start date less than end date in number_of_months'),nl,fail

Prolog_library - 8
number_of_years_from_to(Date_from,Date_to,N):-
    year(Date_to,Yr_to),
    year(Date_from,Yr_from),
    month(Date_to,Mo_to),
    month(Date_from,Mo_from),
    ( Mo_to >= Mo_from, 
    N is (Yr_to - Yr_from) 
    ;
    Mo_to < Mo_from, 
    N is (Yr_to - Yr_from - 1) 
    ),
    !,
    (N<0,write('Start date less than end date in number_of_years'),nl,fail
    ;true).

done(library).
Index to Procedures in Prolog Library

ancestor_list(Hierarchy,Descendant_node, L) ...................... 6
append([X|L1],[L2,[X|L3]]) ......................................... 2
avg(X,G,_) .................................................................. 3
avg(X,G,A) .................................................................. 3
avg_found(I,J,K) .......................................................... 3
avg_found(M,N,A) .......................................................... 3
cardinality([H|T],C) ....................................................... 2
cardinality([T],C) .......................................................... 2
cnt(X,G,_) .................................................................. 3
cnt(X,G,C) .................................................................. 3
cnt_found(I,J) ................................................................. 3
date_function(D,Y,F,Target_date,Function_value) .............. 6
date_function_long_way(D,Y,F,Target_date,Function_value) .... 6
day(Date,Da) .................................................................. 7
fast_setof(Template,Goal,Set) .......................................... 3
get_fast_set(Set) ............................................................... 4
getnext(X) ..................................................................... 3
intersection([X|R],Y,Z) ..................................................... 2
intersection([X|R],Y,[X|Z]) ................................................. 2
is_a(Hierarchy,Descendant,Ancestor) ................................. 5, 6
max(X,G,_) .................................................................. 3
max(X,G,M) .................................................................. 3
max_found(I,J) ................................................................ 3
member(X,[Y|Y]) .............................................................. 2
min(X,G,_) .................................................................. 3
min(X,G,M) .................................................................. 3
min_found(I,J) ................................................................ 3
month(Date,Mo) ................................................................ 8
number_of_days_from_to(Date_from,Date_to,N) ................. 8
number_of_months_from_to(Date_from,Date_to,N) ............... 8
number_of_years_from_to(Date_from,Date_to,N) ................. 9
once(X) ........................................................................ 2
put_list([H | Remainder_list],L,Length,Chars_done) ............ 5
put_spaces(Number_of_spaces) ............................................ 5
r(X) ............................................................................ 4
retractall(X) ................................................................. 2
round_positive(X,to N decimal_places(X,N,Rounded_X)). ....... 5
round(X_to N dec_places(X,N,Rounded_X)) ......................... 4
round(X_to two dec_places(X,Rounded_X)) ......................... 4
set_diff([A|X],[B,C]) ...................................................... 2
set_diff([A|X],[B|A|C]) .................................................... 2
subset([A|X],[Y]) .......................................................... 2
sum(X,G,_) .................................................................. 3
sum(X,G,S) .................................................................. 3
sum([H|T],S) ................................................................ 3
sum([T],S) .................................................................. 2
sum_found(I,J) ............................................................... 3
union([X|R],Y,Z) ............................................................ 2
union([X|R],Y,[X|Z]) ....................................................... 2
w(N) ............................................................................ 5
write_down(X,G) .............................................................. 3
write_fixed_length(Field_name,Length) .............................. 5
year(Date,Yr) .................................................................. 8

Prolog_library - 10
APPENDIX 5: TRADING DATABASE

This Appendix contains 929 lines (20 pages) of Prolog facts (equivalent to relational database tables) for the Interpreter. Tables have been organized into logically-related groups as follows:

<table>
<thead>
<tr>
<th>Tables</th>
<th>pages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formula Accounting</strong></td>
<td></td>
</tr>
<tr>
<td>This section contains tables for the Formula Accounting system, as defined in Chapter 8, Figure 8.8 except for those created by the Interpreter. All these tables are headed Table 1.xx.</td>
<td>2-7</td>
</tr>
<tr>
<td><strong>Resources and Exchange Events Model</strong></td>
<td></td>
</tr>
<tr>
<td>This section contains tables for the REE Accounting model, as defined in Chapter 9, Figure 9.9. All these tables are headed Table 2.xx or Table 3.xx.</td>
<td>7-15</td>
</tr>
<tr>
<td>This section contains tables for the Trading Exchange Events from Chapter 9, Table 9.1. eh(..) and rc(..) predicates have been interleaved to match Table 9.1.</td>
<td>15</td>
</tr>
<tr>
<td><strong>Interpreter</strong></td>
<td></td>
</tr>
<tr>
<td>This section contains tables that are not part of either the FA system or for the REE model, but which are required by the Interpreter, e.g., tables to map from one model to the other, from accounting concepts to GL accounts, and to invoke the correct procedure to process an Exchange Event. All these tables are headed Table 4.xx.</td>
<td>15-21</td>
</tr>
</tbody>
</table>

There is an index to all tables in this Appendix on page 22.
/*---------------------------------------------------------*/
/* Table 1.1: chart_of_accounts/GLacno,description,class. */
/*---------------------------------------------------------*/

date_of_accounts(11101,'Petty Cash','Cash').
date_of_accounts(11102,'Cash at Bank','Cash').
date_of_accounts(11103,'Bank 1','Cash').
date_of_accounts(11201,'Customer Excel Steel','Accounts Receivable - Gross').
date_of_accounts(11202,'Customer Farm Products','Accounts Receivable - Gross').
date_of_accounts(11203,'Customer L. Good','Accounts Receivable - Gross').
date_of_accounts(11204,'Customer Hardware Manufacturing','Accounts Receivable - Gross').
date_of_accounts(11205,'Customer Instruments Ltd','Accounts Receivable - Gross').
date_of_accounts(11206,'Customer Jones Auto Repairs','Accounts Receivable - Gross').
date_of_accounts(11207,'Customer Kirk and Nobby','Accounts Receivable - Gross').
date_of_accounts(11208,'Customer 1','Accounts Receivable - Gross').
date_of_accounts(11100,'Allowance for Doubtful Debts','Allowance for Doubtful Debts').
date_of_accounts(11401,'AL350,125mm Maxco Bench Lathe','Trading Inventory').
date_of_accounts(11402,'AL900,153mm Maxco Bench Lathe','Trading Inventory').
date_of_accounts(11403,'TL640,166mm Maxco Bench Lathe','Trading Inventory').
date_of_accounts(11404,'1000G,419mm Maxco Geared Head Lathe','Trading Inventory').
date_of_accounts(11405,'C1250,Lathe','Trading Inventory').
date_of_accounts(11406,'350SSteel Stand','Trading Inventory').
date_of_accounts(11407,'900SSteel Stand','Trading Inventory').
date_of_accounts(11408,'640SSteel Stand','Trading Inventory').
date_of_accounts(11409,'M1Parkson Motor, 1HP, 240 volt','Trading Inventory').
date_of_accounts(11410,'M2Parkson Motor, 1.5HP, 240 volt','Trading Inventory').
date_of_accounts(11411,'M3Parkson Motor, 1HP, 415 volt','Trading Inventory').
date_of_accounts(11412,'PL200Parkson Coolant Pump, 415 volt','Trading Inventory').
date_of_accounts(11413,'Widget 1','Trading Inventory').
date_of_accounts(11414,'Widget 2','Trading Inventory').
date_of_accounts(11500,'Store Supplies','Store Supplies').
date_of_accounts(11600,'Prepaid Advertising','Prepayments').
date_of_accounts(11700,'Prepaid Insurance','Prepayments').
date_of_accounts(11800,'Prepaid Wages','Prepayments').
date_of_accounts(12100,'Land','Land').
date_of_accounts(12200,'Building','Building - Gross').
date_of_accounts(12300,'Accumulated Depreciation - Building','Accumulated Depreciation - Building').
date_of_accounts(12401,'Office Equipment','Office Equipment - Gross').
date_of_accounts(12402,'Champion Deluxe Electric Typewriter - Gross','Office Equipment - Gross').
date_of_accounts(12403,'Typhoon Electric Fan - Gross','Office Equipment - Gross').
date_of_accounts(12501,'Accumulated Depreciation - Office Equipment','Accumulated Depreciation - Office Equipment').
date_of_accounts(12502,'Accumulated Depreciation - Champion Typewriter','Accumulated Depreciation - Office Equipment').
date_of_accounts(12503,'Accumulated Depreciation - Typhoon Fan','Accumulated Depreciation - Office Equipment').
date_of_accounts(12600,'Store Equipment','Store Equipment - Gross').
date_of_accounts(12601,'Electric Hoist','Store Equipment - Gross').
date_of_accounts(12602,'Qualift Electric Transporter','Store Equipment - Gross').
date_of_accounts(12700,'Accumulated Depreciation - Store Equipment','Accumulated Depreciation - Store Equipment').

Trading_database - 2
chart_of_accounts(12701,'Accumulated Depreciation - Electric Hoist','Accumulated Depreciation - Store Equipment').
chart_of_accounts(12702,'Accumulated Depreciation - Qualift Transporter','Accumulated Depreciation - Store Equipment').
chart_of_accounts(12801,'1990 Ford Delivery Truck','Vehicles - Gross').
chart_of_accounts(12901,'AccumDepr - 1990 Ford Truck','Accumulated Depreciation - Vehicles').
chart_of_accounts(21101,'Newman Imports','Accounts Payable').
chart_of_accounts(21102,'O.K. Tools','Accounts Payable').
chart_of_accounts(21103,'Park Electrical','Accounts Payable').
chart_of_accounts(21104,'Quality Distributors','Accounts Payable').
chart_of_accounts(21105,'Reliable Office Supplies','Accounts Payable').
chart_of_accounts(21106,'Stands Ltd.','Accounts Payable').
chart_of_accounts(21107,'Supplier 1','Accounts Payable').
chart_of_accounts(21108,'Supplier 2','Accounts Payable').
chart_of_accounts(21201,'Office Wages Payable','Wages Payable').
chart_of_accounts(21202,'Sales Wages Payable','Wages Payable').
chart_of_accounts(21300,'Sales Tax Payable','Accruals').
chart_of_accounts(21400,'Delivery Charges Payable','Accruals').
chart_of_accounts(21502,'Dividend Payable - SH1','Dividend Payable').
chart_of_accounts(21503,'Dividend Payable - SH2','Dividend Payable').
chart_of_accounts(22100,'Mortgage','Mortgage').
chart_of_accounts(31100,'A. Mackintosh - Capital','Capital').
chart_of_accounts(31101,'Issued Capital','Capital').
chart_of_accounts(31102,'Share Premium Reserve','Capital').
chart_of_accounts(31103,'Share Discount','Capital').
chart_of_accounts(31200,'A. Mackintosh - Drawings','Capital').
chart_of_accounts(31300,'Opening Balance Fudge','Retained Earnings').
chart_of_accounts(31400,'Prior Period Gearing Gain','Retained Earnings').
chart_of_accounts(31401,'Gearing Gain','Revenue').
chart_of_accounts(31500,'Prior Period CCA Gain above General Prices','Retained Earnings').
chart_of_accounts(31501,'CCA Gain above General Prices','Revenue').
chart_of_accounts(31600,'Prior Period CCA Standard Cost Interindex Gain','Retained Earnings').
chart_of_accounts(31601,'CCA Standard Cost Interindex Gain','Revenue').
chart_of_accounts(31700,'Prior Period Exchange Rate Fluctuation','Retained Earnings').
chart_of_accounts(31701,'Exchange Rate Fluctuation','Revenue').
chart_of_accounts(31800,'Retained Earnings','Retained Earnings').
chart_of_accounts(31900,'Profit and Loss','Retained Earnings').
chart_of_accounts(32500,'Non-operating Income','Other Non-operating Income').
chart_of_accounts(32600,'Dividend Received','Dividend Received').
chart_of_accounts(32700,'Abnormal Income/Expense','Abnormal Income/Expense').
chart_of_accounts(32100,'Sales Revenue','Revenue').
chart_of_accounts(32200,'Discount Expense','Revenue').
chart_of_accounts(32300,'Sales Returns and Allowances','Revenue').
chart_of_accounts(32400,'Bad and Doubtful Debts','Revenue').
chart_of_accounts(33100,'Cost of Sales','Cost of Sales').
chart_of_accounts(33200,'Purchase Discounts','Cost of Sales').
chart_of_accounts(33301,'Sales Wages Expense','Selling Expense').
chart_of_accounts(33302,'Advertising Expense','Selling Expense').

Trading_database - 3
chart_of_accounts(33303,'Store Supplies Expense','Selling Expense').
chart_of_accounts(33304,'Rent Expense','Selling Expense').
chart_of_accounts(33305,'Depreciation Expense - Store Equipment','Selling Expense').
chart_of_accounts(33306,'Miscellaneous Selling Expense','Selling Expense').
chart_of_accounts(33401,'Office Wages Expense','General Expense').
chart_of_accounts(33402,'Utilities Expense','General Expense').
chart_of_accounts(33403,'Depreciation Expense - Office Equipment','General Expense').
chart_of_accounts(33404,'Depreciation Expense - Building','General Expense').
chart_of_accounts(33405,'Stock Losses','General Expense').
chart_of_accounts(33406,'Miscellaneous General Expense','General Expense').
chart_of_accounts(33407,'Depreciation Expense - 1990 Ford Truck','General Expense').
chart_of_accounts(33408,'Donation Expense','General Expense').
chart_of_accounts(33501,'Interest Expense','Financial Expense').

/*chart_of_accounts(34101,'Profit and Loss','Profit and Loss'). */

/* Table 1.2: gIAC_hierarchy(P_seq,P_name,C_seq,C_name). */

117  gIAC_hierarchy(0,'Account',1,'Asset').
118  gIAC_hierarchy(0,'Account',3,'Owners Equity'). /* deliberately out of seq to test reporting */
119  gIAC_hierarchy(0,'Account',2,'Liability').
120  gIAC_hierarchy(1,'Asset',1,'Current Asset').
121  gIAC_hierarchy(1,'Asset',2,'Fixed Asset').
122  gIAC_hierarchy(1,'Asset',3,'Other Asset').
123  gIAC_hierarchy(1,'Liability',1,'Current Liability').
124  gIAC_hierarchy(1,'Liability',2,'Long Term Liability').
125  gIAC_hierarchy(1,'Owners Equity',0,'Issued Capital').
126  gIAC_hierarchy(1,'Owners Equity',1,'Capital').
127  gIAC_hierarchy(1,'Owners Equity',2,'Retained Earnings').
128  gIAC_hierarchy(1,'Owners Equity',3,'Share Premium Reserve').
129  gIAC_hierarchy(1,'Owners Equity',4,'Asset Revaluation').
130  gIAC_hierarchy(1,'Owners Equity',5,'Operating Income').
131  gIAC_hierarchy(1,'Owners Equity',6,'Non-operating Income').
132  /*gIAC_hierarchy(1,'Owners Equity',7,'Profit and Loss'). */
133  gIAC_hierarchy(2,'Current Asset',1,'Cash').
134  gIAC_hierarchy(2,'Current Asset',2,'Accounts Receivable').
135  gIAC_hierarchy(2,'Current Asset',3,'Inventory').
136  gIAC_hierarchy(2,'Current Asset',4,'Prepayments').
137  gIAC_hierarchy(2,'Fixed Asset',1,'Land').
138  gIAC_hierarchy(2,'Fixed Asset',2,'Building').
139  gIAC_hierarchy(2,'Fixed Asset',3,'Plant').
140  gIAC_hierarchy(2,'Fixed Asset',4,'Vehicles').
141  gIAC_hierarchy(2,'Fixed Asset',5,'Office Equipment').
142  gIAC_hierarchy(2,'Fixed Asset',6,'Store Equipment').
143  gIAC_hierarchy(2,'Other Asset',1,'Goodwill').
144  gIAC_hierarchy(2,'Long Term Liability',1,'Debenture').
145  gIAC_hierarchy(2,'Long Term Liability',2,'Lease').
gIAC_hierarchy(2,'Long Term Liability',3,'Mortgage').
gIAC_hierarchy(2,'Operating Income',1,'Gross Profit').
gIAC_hierarchy(2,'Operating Income',2,'Expense').
gIAC_hierarchy(2,'Non-operating Income',1,'Dividend Received').
gIAC_hierarchy(2,'Non-operating Income',2,'Abnormal Income/Expense').
gIAC_hierarchy(2,'Non-operating Income',3,'Other Non-operating Income').
gIAC_hierarchy(3,'Gross Profit',1,'Revenue').
gIAC_hierarchy(3,'Gross Profit',2,'Cost of Sales').
gIAC_hierarchy(3,'Accounts Receivable',1,'Accounts Receivable - Gross').
gIAC_hierarchy(3,'Accounts Receivable',2,'Allowance for Doubtful Debts').
gIAC_hierarchy(3,'Inventory',1,'Raw Materials Inventory').
gIAC_hierarchy(3,'Inventory',2,'WIP Inventory').
gIAC_hierarchy(3,'Inventory',3,'Finished Goods Inventory').
gIAC_hierarchy(3,'Inventory',4,'Trading Inventory').
gIAC_hierarchy(3,'Inventory',5,'Store Supplies').
gIAC_hierarchy(3,'Building',1,'Building - Gross').
gIAC_hierarchy(3,'Building',2,'Accumulated Depreciation - Building').
gIAC_hierarchy(3,'Plant',1,'Plant - Gross').
gIAC_hierarchy(3,'Plant',2,'Accumulated Depreciation - Plant').
gIAC_hierarchy(3,'Vehicles',1,'Vehicles - Gross').
gIAC_hierarchy(3,'Vehicles',2,'Accumulated Depreciation - Vehicles').
gIAC_hierarchy(3,'Office Equipment',1,'Office Equipment - Gross').
gIAC_hierarchy(3,'Office Equipment',2,'Accumulated Depreciation - Office Equipment').
gIAC_hierarchy(3,'Store Equipment',1,'Store Equipment - Gross').
gIAC_hierarchy(3,'Store Equipment',2,'Accumulated Depreciation - Store Equipment').
gIAC_hierarchy(3,'Current Liability',1,'Accounts Payable').
gIAC_hierarchy(3,'Current Liability',2,'Tax Payable').
gIAC_hierarchy(3,'Current Liability',3,'Dividend Payable').
gIAC_hierarchy(3,'Current Liability',4,'Wages Payable').
gIAC_hierarchy(3,'Current Liability',5,'Labour Clearing').
gIAC_hierarchy(3,'Current Liability',6,'Accruals').
gIAC_hierarchy(3,'Expense',1,'Rent Expense').
gIAC_hierarchy(3,'Expense',2,'Interest Expense').
gIAC_hierarchy(3,'Expense',3,'Advertising Expense').
gIAC_hierarchy(3,'Expense',4,'Discount Expense').
gIAC_hierarchy(3,'Expense',5,'Amortization Expense').
gIAC_hierarchy(3,'Expense',6,'Lease Expense').
gIAC_hierarchy(3,'Expense',7,'Standard Cost Variance').
gIAC_hierarchy(3,'Expense',8,'Production OH').
gIAC_hierarchy(3,'Expense',9,'Marketing Expense').
gIAC_hierarchy(3,'Expense',10,'Despatch Expense').

Trading_database - 5
gl AC hierarchy(3,'Expense',11,'Selling Expense').
gl AC hierarchy(3,'Expense',12,'General Expense').
gl AC hierarchy(3,'Expense',13,'Financial Expense').

/*
 * Table 1.3 jeh(JEno,Date,Eno,Narration).
 * generated by Interpreter */
*/

/*
 * Table 1.4 jed(JEno,Line_no,JE_ACno,Amount)
 * generated by Interpreter */
*/

/*
 * Table 1.5 gl depreciation(Asset_Acno,Accum_depr_Acno,Depr_exp_Acno).
 */

gl depreciation(12200,12300,33404).
gl depreciation(12401,12501,33403).
gl depreciation(12402,12502,33403).
gl depreciation(12403,12503,33403).
gl depreciation(12600,12700,33305).
gl depreciation(12601,12701,33305).
gl depreciation(12602,12702,33305).
gl depreciation(12801,12901,33307).

gl amortization(11700, 33406).
gl amortization(11800, 33401).
/*gl amortization(11900, 33304). */
/*gl amortization(11901, 33304). */

/*
 * Table 1.8 expense ac for growing liability(GL_Acno,Expense_Acno).
 */
expen ac for growing liability(22100,33304)./* Mortgage, not used*/

/*
 * Table 1.9 wip_GLac_class(GL_Acno,WIP_class). (Prime costs only).
 */

/*
 * Table 1.10 allocation of OH(From_GLac,To_GLac,Proportion).
 */

/* rent */
allocation of OH(2702,2810,0.6).
allocation of OH(2702,2820,0.3).
allocation of OH(2702,2830,0.1).

/* car lease expense */
allocation of OH(2706,2810,0.5).
allocation of OH(2706,2820,0.5).

/* general factory overhead */
allocation of OH(2802,2810,0.7).

Trading database - 6
230 allocation_of_OH(2802,2820,0.3).
231 /*----------------------------------------------*/
232 /* Table 1.1 department_GL_variance_account(Department, Variance_ac_description, GL_Acno). */
233 /*----------------------------------------------*/
234 department_GL_variance_account('Moulding', 'OH Incurred', 2810).
236 department_GL_variance_account('Moulding', 'Material Usage Variance', 2813).
237 department_GL_variance_account('Moulding', 'Labour Usage Variance', 2814).
238 department_GL_variance_account('Moulding', 'OH Spending Variance', 2815).
239 department_GL_variance_account('Moulding', 'OH Lab Efficiency Variance', 2816).
240 department_GL_variance_account('Moulding', 'OH Volume Variance', 2817).
241 department_GL_variance_account('Packing', 'OH Incurred', 2820).
243 department_GL_variance_account('Packing', 'Material Usage Variance', 2823).
244 department_GL_variance_account('Packing', 'Labour Usage Variance', 2824).
245 department_GL_variance_account('Packing', 'OH Spending Variance', 2825).
246 department_GL_variance_account('Packing', 'OH Lab Efficiency Variance', 2826).
247 department_GL_variance_account('Packing', 'OH Volume Variance', 2827).

248 /*----------------------------------------------*/
249 /* Table 2.1 ao_account(ACno, Description, Class, Rno, Agent_id, Controller). */
250 /*----------------------------------------------*/
251 ao_account(11101, 'PettyCash', 'Cash', 1001, agent, controller).
252 ao_account(11102, 'Cash at Bank', 'Cash', 1002, agent, controller).
253 ao_account(11103, 'Bank 1', 'Cash', 1002, agent, controller).
254 ao_account(11201, 'CustomerExcel Steel', 'Accounts Receivable', 1012, 'C01', controller).
255 ao_account(11202, 'CustomerFarm Products', 'Accounts Receivable', 1012, 'C02', controller).
256 ao_account(11203, 'Customer L. Good', 'Accounts Receivable', 1012, 'C03', controller).
258 ao_account(11205, 'Customer Instruments Ltd', 'Accounts Receivable', 1012, 'C05', controller).
259 ao_account(11206, 'Customer Jones Auto Repairs', 'Accounts Receivable', 1012, 'C06', controller).
262 ao_account(11401, 'AL350, 125mm Maxco Bench Lathe', 'Inventory', 1020, agent, controller).
263 ao_account(11402, 'AL900, 153mm Maxco Bench Lathe', 'Inventory', 1021, agent, controller).
264 ao_account(11403, 'HL640, 166mm Maxco Bench Lathe', 'Inventory', 1022, agent, controller).
265 ao_account(11404, '1000G, 419mm Maxco Geared Head Lathe', 'Inventory', 1023, agent, controller).
266 ao_account(11405, 'C1250, Lathe', 'Inventory', 1024, agent, controller).
267 ao_account(11406, '350S Steel Stand', 'Inventory', 1025, agent, controller).
268 ao_account(11407, '500S Steel Stand', 'Inventory', 1026, agent, controller).
269 ao_account(11408, '640S Steel Stand', 'Inventory', 1027, agent, controller).
270 ao_account(11409, 'M1 Parkson Motor, 1HP, 240 volt', 'Inventory', 1028, agent, controller).
271 ao_account(11410, 'M2 Parkson Motor, 1.5HP, 240 volt', 'Inventory', 1029, agent, controller).
272 ao_account(11411, 'M3 Parkson Motor, 1HP, 415 volt', 'Inventory', 1030, agent, controller).
273 ao_account(11412, 'PL200 Parkson Coolant Pump, 415 volt', 'Inventory', 1031, agent, controller).
274 ao_account(11413, 'Widget 1', 'Inventory', 1032, agent, controller).
275 ao_account(11414, 'Widget 2', 'Inventory', 1033, agent, controller).
276 ao_account(11420, 'RMInventory 1', 'Inventory', 1034, agent, controller).
277 ao_account(11421, 'RMInventory 2', 'Inventory', 1035, agent, controller).
278 ao_account(11430, 'Unpacked IL bottles', 'Inventory', 1035, agent, controller).

Trading_database - 7
ao_account(11500,'Store Supplies','Store Supplies',1040,agent,controller).
ao_account(11501,'Office Supplies','Office Supplies',1041,agent,controller).
ao_account(11502,'Store Supplies Services','Store Supplies Services',1512,agent,controller).
ao_account(11600,'Prepaid Advertising','Prepaid Advertising',1012,agent,controller).
ao_account(11601,'Customer Goodwill','Customer Goodwill',1502,agent,controller).
ao_account(11602,'Sales Labour Services','Labour Services',1503,agent,controller).
ao_account(11603,'Advertising Services','Advertising',1500,agent,controller).
ao_account(11700,'Insurance Services','Insurance',1507,agent,controller).
ao_account(11701,'Office Labour Services','Labour Services',1505,agent,controller).
ao_account(11702,'Banking Services','Banking Services',1511,agent,controller).
ao_account(11900,'Factory Occupancy Rights','Occupancy Rights',1501,agent,controller).
ao_account(11901,'Water Usage Rights','Occupancy Rights',1508,agent,controller).
ao_account(11902,'Telephone Services','Occupancy Rights',1509,agent,controller).
ao_account(11903,'Electricity Services','Occupancy Rights',1510,agent,controller).
ao_account(12000,'Delivery Services','Delivery Services',1506,agent,controller).
ao_account(12100,'Land','Land',1104,agent,controller).
ao_account(12200,'Building','Buildings',1100,agent,controller).
ao_account(12401,'Office Equipment','Office Equipment',1101,agent,controller).
ao_account(12402,'Champion Deluxe Electric Typewriter','Office Equipment',1103,agent,controller).
ao_account(12403,'Typhoon Electric Fan','Office Equipment',1107,agent,controller).
ao_account(12600,'Store Equipment','Store Equipment',1102,agent,controller).
ao_account(12601,'Electric Hoist','Store Equipment',1105,agent,controller).
ao_account(12602,'Qualift Electric Transporter','Store Equipment',1106,agent,controller).
ao_account(12801,'1990 Ford Delivery Truck','Vehicles',1108,agent,controller).
ao_account(21101,'Newman Imports','Accounts Payable',1012,'S01',controller).
ao_account(21102,'O.K. Tools','Accounts Payable',1012,'S02',controller).
ao_account(21103,'Park Electrical','Accounts Payable',1012,'S03',controller).
ao_account(21104,'Quality Distributors','Accounts Payable',1012,'S04',controller).
ao_account(21105,'Reliable Office Supplies','Accounts Payable',1012,'S05',controller).
ao_account(21106,'Stands Ltd.','Accounts Payable',1012,'S06',controller).
ao_account(21107,'Supplier 1','Accounts Payable',1012,'S06',controller).
ao_account(21108,'Supplier 2','Accounts Payable',1012,'S06',controller).
ao_account(21200,'Wages Payable','Wages Payable',1012,'many employees',controller).
ao_account(21300,'Sales Tax Payable','Sales Tax Payable',1012,'Sales Tax Office',controller).
ao_account(21400,'Delivery Charges Payable','Delivery Charges Payable',1012,'S07',controller).
ao_account(21502,'Dividend to Shareholder 1','Dividend Payable',1012,'S07',controller).
ao_account(21503,'Dividend to Shareholder 2','Dividend Payable',1012,'S07',controller).
ao_account(21504,'Dividend to Shareholder 3','Dividend Payable',1012,'S07',controller).
ao_account(21505,'Dividend to Shareholder 4','Dividend Payable',1012,'S07',controller).
ao_account(31100,'A. Mackintosh - Owner','Owners" Equity',1402,agent,controller).
ao_account(31101,'A. Mackintosh - Drawings','Owners" Equity',1401,agent,controller).
ao_account(31102,'Stock Holder 1','Owners" Equity',1403,agent,controller).
ao_account(31103,'Stock Holder 2','Owners" Equity',1403,agent,controller).
ao_account(31202,'Stock Holder 1 - Component','Owners" Equity',1404,agent,controller).
ao_account(31203,'Stock Holder 2 - Component','Owners" Equity',1404,agent,controller).

/*----------------------------------*/
/* Table 2.2 ao_hierarchy(Pl,Pname,Cseq,Cname).*/

Trading_database - 8
ao_hierarchy(0,'Asset or Obligation Account',1,'Asset').
ao_hierarchy(0,'Asset or Obligation Account',2,'Obligation').
ao_hierarchy(1,'Asset',1,'Unconditional Asset').
ao_hierarchy(1,'Asset',2,'Conditional Asset').
ao_hierarchy(1,'Obligation',1,'Unconditional Obligation').
ao_hierarchy(1,'Obligation',2,'Conditional Obligation').
ao_hierarchy(2,'Unconditional Asset',1,'Current Asset').
ao_hierarchy(2,'Unconditional Asset',2,'Fixed Asset').
ao_hierarchy(2,'Unconditional Asset',3,'Other Asset').
ao_hierarchy(2,'Unconditional Obligation',1,'Current Liability').
ao_hierarchy(2,'Unconditional Obligation',2,'Long Term Liability').
ao_hierarchy(2,'Unconditional Obligation',3,'Owners' Equity').
ao_hierarchy(2,'Conditional Asset',1,'Sales Order').
ao_hierarchy(2,'Conditional Asset',2,'Goods on Order').
ao_hierarchy(2,'Conditional Asset',3,'Planned Production').
ao_hierarchy(2,'Conditional Obligation',1,'Purchase Order').
ao_hierarchy(2,'Conditional Obligation',2,'Goods Promised').
ao_hierarchy(2,'Conditional Obligation',3,'Resources Committed').
ao_hierarchy(3,'Current Asset',1,'Cash').
ao_hierarchy(3,'Current Asset',2,'Accounts Receivable').
ao_hierarchy(3,'Current Asset',3,'Inventory').
ao_hierarchy(3,'Current Asset',4,'Occupancy Rights').
ao_hierarchy(3,'Current Asset',5,'Advertising').
ao_hierarchy(3,'Current Asset',6,'Delivery Services').
ao_hierarchy(3,'Current Asset',7,'Insurance Services').
ao_hierarchy(3,'Current Asset',8,'Idle Time').
ao_hierarchy(3,'Current Asset',9,'Customer Goodwill').
ao_hierarchy(3,'Current Asset',10,'Labour Services').
aoHierarchy(3,'Current Asset',11,'Banking Services').
aoHierarchy(3,'Current Asset',12,'Store Supplies Services').
aoHierarchy(3,'Fixed Asset',1,'Investment').
aoHierarchy(3,'Fixed Asset',2,'Plant').
aoHierarchy(3,'Fixed Asset',3,'Vehicles').
aoHierarchy(3,'Fixed Asset',4,'Land').
aoHierarchy(3,'Fixed Asset',5,'Buildings').
aoHierarchy(3,'Fixed Asset',6,'Office Equipment').
aoHierarchy(3,'Fixed Asset',7,'Store Equipment').
aoHierarchy(3,'Other Asset',1,'Goodwill').
aoHierarchy(3,'Current Liability',1,'Accounts Payable').
aoHierarchy(3,'Current Liability',2,'Wages Payable').
aoHierarchy(3,'Current Liability',3,'Tax Payable').
aoHierarchy(3,'Current Liability',4,'Sales Tax Payable').
aoHierarchy(3,'Current Liability',5,'Dividend Payable').
aoHierarchy(3,'Current Liability',6,'Interest Payable').
aoHierarchy(3,'Current Liability',7,'Delivery Charges Payable').
aoHierarchy(3,'Current Liability',8,'Warranty').
aoHierarchy(3,'Current Liability',9,'Wages Payable (in hours)').
375  ao_hierarchy(3,'Long Term Liability',1,'Debentures').
376  ao_hierarchy(3,'Long Term Liability',2,'Lease Obligation').
377  ao_hierarchy(3,'Long Term Liability',3,'Mortgage Loan').
378  ao_hierarchy(4,'Inventory',1,'Store Supplies').
379  ao_hierarchy(4,'Inventory',2,'Office Supplies').
380  /* Table 2.3 resource(Rno,Description,Unit_of_measure,Rtype). */
381  */
382  resource(1001,'Cash on Hand','$','Cash').
383  resource(1002,'Cash on Hand','Yen','Cash at Bank, Current A/C').
384  resource(1003,'Cash on Hand','Yen','Cash at Bank, Interest Earning').
385  resource(1004,'Cash on Hand','Yen','Foreign Currency').
386  resource(1005,'Cash on Hand','$US','Foreign Currency').
387  resource(1006,'Cash on Hand','Pound','Foreign Currency').
388  resource(1007,'Cash on Hand','Dmark','Foreign Currency').
389  resource(1008,'Immediate Debt','$','Other Debt').
390  resource(1009,'7 Day Debt','$','Trade Debt').
391  resource(1010,'30 Day Debt','$','Trade Debt').
392  resource(1011,'30 to 90 Day Debt','$','Other Debt').
393  resource(1012,'Foreign Debt','Yen','Foreign Currency Trade Debt').
394  resource(1013,'Foreign Debt','$US','Foreign Currency Trade Debt').
395  resource(1014,'Foreign Debt','Pound','Foreign Currency Trade Debt').
396  resource(1015,'Foreign Debt','Dmark','Foreign Currency Trade Debt').
397  resource(1016,'Foreign Debt','Yen','Foreign Currency Trade Debt').
398  resource(1017,'Foreign Debt','$US','Foreign Currency Trade Debt').
399  resource(1018,'Foreign Debt','Pound','Foreign Currency Trade Debt').
400  resource(1019,'Foreign Debt','Dmark','Foreign Currency Trade Debt').
401  resource(1020,'AL350125 mm Maxco Bench Lathe','unit','Inventory').
402  resource(1021,'AL900153 mm Maxco Bench Lathe','unit','Inventory').
403  resource(1022,'HL640 166 mm Maxco Bench Lathe','unit','Inventory').
404  resource(1023,'1000G419 mm Maxco Geared Head Lathe','unit','Inventory').
405  resource(1024,'M1 Parkson Motor 1 H.P. 240 volt','unit','Inventory').
406  resource(1025,'M2 Parkson Motor 1.5 H.P. 240 volt','unit','Inventory').
407  resource(1026,'M3 Parkson Motor 1 H.P. 415 volt','unit','Inventory').
408  resource(1027,'PL200 Parkson Coolant Pump','unit','Inventory').
409  resource(1028,'Widget 1','unit','Inventory').
410  resource(1029,'Widget 2','unit','Inventory').
411  resource(1030,'1990Ford Delivery Truck','unit','Vehicle').
412  resource(1031,'5% Debenture, 5 years (in dollars)','Debenture').
413  resource(1032,'7% Debenture, 5 years (in dollars)','Debenture').
414  */
415  */
416  */
417  */
418  */
419  */
420  */
421  */
422  */
423  */
424  */
425  Trading_database - 10
resource(1302,'14% Mortgage Loan, x years','$1 loan','Loan').
resource(1400,'Ordinary Share (stock) in self','Unit','Share (stock)').
resource(1401,'Share (stock) in R','Unit','Share (stock)').
resource(1402,'Proprietorship','$1 unit','Proprietorship Unit').
resource(1403,'Share, Par value $2','unit','Share (stock)').
resource(1404,'Right to dividend','unit','Share (stock)').

resource(1500,'Advertising','$','Informed Customer').
resource(1501,'Factory Occupancy Rights','Month','Occupancy Rights').
resource(1502,'Customer Goodwill','$','Customer Goodwill').

resource(1503,'Sales Labour Services',fortnight,'Labour Services').
resource(1504,'Office Occupancy Rights','Month','Labour Services').
resource(1505,'Office Labour Services',fortnight,'Labour Services').
resource(1506,'Delivery Services','Delivery','Delivery Services').
resource(1507,'Insurance Services','month','Insurance Services').
resource(1508,'Water Usage Rights','$','Occupancy Rights').
resource(1509,'Telephone Services','$','Occupancy Rights').
resource(1510,'Electricity Services','$','Occupancy Rights').
resource(1511,'Banking Services','$','Banking Services').
resource(1512,'Store Supplies Services','$','Store Supplies Services').

resource(1601,'Car Lease Receipts Car ABD456','Lease','Lease').
resource(1602,'Car Lease Receipts Car ABD457','Lease','Lease').
resource(1701,'Warranty Protection - 2L bottles','Bottle','Warranty').

/*-----------------------------*/
/*  Table 2.4A resource_hierarchy(Pl,Pname,Cseq,Cname).*/
resource_hierarchy(0,'Resource',1,'Monetary Resource').
resource_hierarchy(0,'Resource',2,'Non-monetary (N.M.) Resource').
resource_hierarchy(1,'Monetary Resource',1,'Cash Available Immediately').
resource_hierarchy(1,'Monetary Resource',2,'Single Future Flow of Money').
resource_hierarchy(1,'Monetary Resource',3,'Stream of Future Flows of Money').
resource_hierarchy(1,'Non-monetary Resource',1,'Tangible Resource').
resource_hierarchy(1,'Non-monetary Resource',2,'Intangible N.M. Resource').
resource_hierarchy(2,'Cash Available Immediately',1,'Cash').
resource_hierarchy(2,'Cash Available Immediately',2,'Cash at Bank, Current A/C').
resource_hierarchy(2,'Cash Available Immediately',3,'Cash at Bank, Interest Earning').
resource_hierarchy(2,'Cash Available Immediately',4,'Foreign Currency').
resource_hierarchy(2,'Single Future Flow of Money',1,'Trade Debt').
resource_hierarchy(2,'Single Future Flow of Money',2,'Other Debt').
resource_hierarchy(2,'Single Future Flow of Money',3,'Foreign Currency Trade Debt').
resource_hierarchy(2,'Stream of Future Flows of Money',1,'Debenture').
resource_hierarchy(2,'Stream of Future Flows of Money',2,'Lease').
resource_hierarchy(2,'Stream of Future Flows of Money',3,'Loan').
resource_hierarchy(2,'Tangible Resource',1,'Identified Tangible Resource').
resource_hierarchy(2,'Tangible Resource',2,'Unidentified Tangible Resource').
resource_hierarchy(2,'Tangible Resource',3,'Unidentified Unclassified Tangible Resource').
resource_hierarchy(2,'Intangible N.M. Resource',1,'Labour Services').
resource_hierarchy(2,'Intangible N.M. Resource',2,'Other Intangible Resource').
resource_hierarchy(3,'Identified Tangible Resource',1,'Raw Materials Inventory (identified)').
resource_hierarchy(3,'Identified Tangible Resource',2,'WIP Inventory (identified)').
resource_hierarchy(3,'Identified Tangible Resource',3,'Finished Goods Inventory (identified)').
resource_hierarchy(3,'Identified Tangible Resource',4,'Land').
resource_hierarchy(3,'Identified Tangible Resource',5,'Amortizable Resource').
resource_hierarchy(3,'Unidentified Tangible Resource',1,'Raw Materials Inventory').
resource_hierarchy(3,'Unidentified Tangible Resource',2,'WIP Inventory').
resource_hierarchy(3,'Unidentified Tangible Resource',3,'Finished Goods Inventory').
resource_hierarchy(3,'Unidentified Tangible Resource',4,'Inventory').
resource_hierarchy(3,'Unidentified Unclassified Tangible Resource',1,'Stores Inventory').
resource_hierarchy(3,'Unidentified Unclassified Tangible Resource',2,'Office Supplies Inventory').
resource_hierarchy(3,'Unidentified Unclassified Tangible Resource',3,'Shipping Equipment').
resource_hierarchy(3,'Other Intangible Resource',1,'Share (stock)').
resource_hierarchy(3,'Other Intangible Resource',2,'Goodwill').
resource_hierarchy(3,'Other Intangible Resource',3,'Inform(ed Customer').
resource_hierarchy(3,'Other Intangible Resource',4,'Machine Services').
resource_hierarchy(3,'Other Intangible Resource',5,'Occupancy Rights').
resource_hierarchy(3,'Other Intangible Resource',6,'Warranty').
resource_hierarchy(3,'Other Intangible Resource',7,'Proprietorship Unit').
resource_hierarchy(3,'Other Intangible Resource',8,'Delivery Services').
resource_hierarchy(3,'Other Intangible Resource',9,'Insurance Services').
resource_hierarchy(3,'Other Intangible Resource',10,'Customer Goodwill').
resource_hierarchy(3,'Other Intangible Resource',11,'Banking Services').
resource_hierarchy(3,'Other Intangible Resource',12,'Store Supplies Services').
resource_hierarchy(4,'Amortizable Resource',1,'Building').
resource_hierarchy(4,'Amortizable Resource',2,'Plant').
resource_hierarchy(4,'Amortizable Resource',3,'Vehicle').
resource_hierarchy(4,'Amortizable Resource',4,'Office Equipment').
resource_hierarchy(4,'Amortizable Resource',5,'Store Equipment').

/*-----------------------------*/
/* Table 2.4B resource_ev_hierarchy(PI,Pname,Cseq,Cname).*/
/* change all this to predictable_future_value(Rno,...) */

/* Negative time derivative */
resource_ev_hierarchy(0,'Resource',1,'Shrinking Value Resource').
resource_ev_hierarchy(0,'Resource',2,'Static Value Resource').
resource_ev_hierarchy(0,'Resource',3,'Growing Value Resource').

/* Zero time derivative resource */
resource_ev_hierarchy(0,'Resource',1,'Shrinking Value Resource').
resource_ev_hierarchy(0,'Resource',2,'Static Value Resource').
resource_ev_hierarchy(0,'Resource',3,'Growing Value Resource').

/* Positive time derivative */
resource_ev_hierarchy(0,'Resource',4,'Variable Time Derivative Resource').

/* Variable time derivative */
resource_ev_hierarchy(1,'Shrinking Value Resource',1,'Tangible Resource').
resource_ev_hierarchy(1,'Shrinking Value Resource',2,'Intangible Resource').
resource_ev_hierarchy(1,'Static Value Resource',1,'Cash').
resource_ev_hierarchy(1,'Static Value Resource',2,'Cash at Bank, Current A/C').
resource_ev_hierarchy(1,'Static Value Resource',3,'Trade Debt').
resource_ev_hierarchy(1,'Static Value Resource',4,'Other Debt').
resource_ev_hierarchy(1,'Static Value Resource',5,'Inventory').
resource_ev_hierarchy(1,'Static Value Resource',6,'Land').

Trading_database - 12
resource_ev_hierarchy(1,'Static Value Resource',8,'Proprietorship Unit').
resource_ev_hierarchy(1,'Growing Value Resource',1,'Cash at Bank, Interest Earning').
resource_ev_hierarchy(1,'Growing Value Resource',2,'Debenture').
resource_ev_hierarchy(1,'Growing Value Resource',3,'Lease').
resource_ev_hierarchy(1,'Growing Value Resource',4,'Loan').
resource_ev_hierarchy(1,'Variable Time Derivative Resource',1,'Farm Animals').
resource_ev_hierarchy(1,'Variable Time Derivative Resource',2,'Foreign Currency').
resource_ev_hierarchy(1,'Variable Time Derivative Resource',3,'Foreign Currency Trade Debt').
resource_ev_hierarchy(2,'Tangible Resource',1,'Building').
resource_ev_hierarchy(2,'Tangible Resource',2,'Plant').
resource_ev_hierarchy(2,'Tangible Resource',3,'Vehicle').
resource_ev_hierarchy(2,'Tangible Resource',4,'Office Equipment').
resource_ev_hierarchy(2,'Tangible Resource',5,'Store Equipment').
resource_ev_hierarchy(2,'Intangible Resource',1,'Labour Services').
resource_ev_hierarchy(2,'Intangible Resource',2,'Occupancy Rights').
resource_ev_hierarchy(2,'Intangible Resource',3,'Machine Services').
resource_ev_hierarchy(2,'Intangible Resource',4,'Warranty').
resource_ev_hierarchy(2,'Intangible Resource',5,'Goodwill').
resource_ev_hierarchy(2,'Intangible Resource',6,'Informed Customer').
resource_ev_hierarchy(2,'Intangible Resource',7,'Delivery Services').
resource_ev_hierarchy(2,'Intangible Resource',8,'Insurance Services').
resource_ev_hierarchy(2,'Intangible Resource',9,'Customer Goodwill').
resource_ev_hierarchy(2,'Intangible Resource',10,'Banking Services').
resource_ev_hierarchy(2,'Intangible Resource',11,'Store Supplies Services').
resource_ev_hierarchy(2,'Inventory',1,'Raw Materials Inventory').
resource_ev_hierarchy(2,'Inventory',2,'WIP Inventory').
resource_ev_hierarchy(2,'Inventory',3,'Finished Goods Inventory').
resource_ev_hierarchy(2,'Inventory',4,'Shipping Equipment Inventory').
resource_ev_hierarchy(2,'Inventory',5,'Stores Inventory').
resource_ev_hierarchy(2,'Inventory',6,'Office Supplies Inventory').

/*---------------------------------------------*/
/*       Table 2.6 agent(Agent_id,Agent_name,Address). */
agent('C01','Excel Steel',27 Bond St., Parraville').
agent('CO2','Farm Products',8 Bay St., Greenwich').
/* etc. */

/*---------------------------------------------*/
/*       Tables 2.7 and 2.8 see at end of REE tables */

/*---------------------------------------------*/
/*       Table 2.9 tied_asset_obligation(ACno1,ACno2).; e.g. leased asset */
/* tied_asset_obligation(104,290). */
/* tied_asset_obligation(106,291). */

/*---------------------------------------------*/
/*       Table 2.10 */
composite_resource(Rno,Component_no,Date_defined,Component_Rno,Qty_of_component).*/

Trading_database - 13
/*composite_resource(Rno,Component_no,Date_defined,Component_Rno,Qty_of_component). */

/*Table 2.11*/
promised_resource(Promised_Rno,Component_no,Date_defined,Component_Rno,Months_from_event_until_control_transfer,Qty_of_component). */
promised_resource(1402,1,860401,1001,0,0).
promised_resource(1302,1,860401,1001,1,410.00).
promised_resource(1302,2,860401,1001,2,410.00).
/*promised_resource(1302,3,860401,1001,3,410.00).etc.*/

/*Table 3.13*/
share_stock(Rno,Par_value,Date_incorporated). */
share_stock(1403,2,891101).

/*Table 3.15*/
expected_value(Rno,Expected_pattern_of_benefits,Expected_life_in_months,Expected_residual_value). */
expected_value(1100,'uniform',240,0).
expected_value(1101,'Reducing balance',29.936).
expected_value(1102,'uniform',120,0).
expected_value(1103,'uniform',120,10).
/*expected_value(1103,'Reducing balance',120,10). */
expected_value(1104,'uniform',120,0).
expected_value(1105,'uniform',120,0).
expected_value(1106,'uniform',120,0).
expected_value(1107,'Reducing balance',120,29.936).
expected_value(1108,'Straight line',120,0).

/* intangible assets should all have zero residual value */
expected_value(1500,'uniform',240,0).
expected_value(1501,'uniform',1,0).
expected_value(1502,'uniform',0,0).
expected_value(1503,'uniform',0,0).
expected_value(1504,'uniform',0,0).
expected_value(1505,'uniform',0,0).
expected_value(1506,'uniform',12,0)./* insurance */
expected_value(1507,'uniform',12,0).
expected_value(1508,'uniform',12,0).
expected_value(1509,'uniform',24,0).

/*Table 3.16*/
growing_resource(Rno,Expected_life_in_months,Annual_growth_rate,Growth_type). */
growing_resource(1003,'No Limit',0.045,'Compound, monthly').
growing_resource(1300,60,0.07,'Simple').
growing_resource(1301,48,0.08,'Simple').
growing_resource(1302,0.14,'Simple').
growing_resource(1601,48,0.12,'Compound, monthly').
growing_resource(1602,48,0.12,'Compound, monthly').
Table 2.7 eh(Eno,Date,Etype,Doc_type,Doc_no,Agent_id,Narrative).

Table 2.8 rc(Eno,Line_no,AOacno,Qty,Value).

eh(1,900101,'Issue Shares',a,b,c,'Issue of 40000 $2 shares at a premium').
rc(1,1,11103,100000_,_).
rc(1,2,113102,-30000_,_).
rc(1,3,113103,-100000_,_).

eh(2,900201,'Credit Purchase',a,b,c,'Purchase of 1000 ea. of Widgets 1 and 2 from Supplier 1').
rc(2,1,11414,1000,7000).
rc(2,2,11414,1000,2000).
rc(2,3,21107,-9000_,_).

eh(3,900301,'Credit Sale',a,b,c,'Sale of 500 Widget 1s to Customer 1').
rc(3,1,11208,50000_,_).
rc(3,2,11413,-500_).

eh(4,900323,'Credit Purchase',a,b,c,'Purchase of 1990 Ford Delivery Truck from Supplier 2').
rc(4,1,12801,13_).
rc(4,2,21108,-36500_,_).

eh(5,900401,'Cash Payment',a,b,c,'Pay Supplier 1 $4,000').
rc(5,1,21107,40000_,_).
rc(5,2,11103,-40000_).

eh(6,900501,'Declare Dividend',a,b,c,'Declaration of dividend of $0.02 per share').
rc(6,1,11202,600_,_).
rc(6,2,11203,200_,_).
rc(6,3,21502,-600_).
rc(6,4,21503,-200_).

eh(7,900601,'Wages Payment',a,b,c,'Pay office wages of $500 for work in May').
rc(7,1,11701,200_,_).
rc(7,2,11103,-500_).

eh(8,900701,'Donation',a,b,c,'Donation to Red Cross').
rc(8,1,11103,-100_).

/*-----------------------------------------------*/
/* Table 4.1 map_AOac_to_GLac(AOac,GLac).* /
/* Mapping from Asset/Obligation Account to General Ledger A/C */

map_AOac_to_GLac(11101,11101).
map_AOac_to_GLac(11102,11102).
map_AOac_to_GLac(11103,11103).
map_AOac_to_GLac(11201,11201).
map_AOac_to_GLac(11202,11202).
map_AOac_to_GLac(11203,11203).
map_AOac_to_GLac(11204,11204).
map_AOac_to_GLac(11205,11205).
map_AOac_to_GLac(11206,11206).
map_AOac_to_GLac(11207,11207).
map_AOac_to_GLac(11208,11208).
map_AOac_to_GLac(11401,11401).
map_AOac_to_GLac(11402,11402).
map_AOac_to_GLac(11403,11403).
map_AOac_to_GLac(11404,11404).
map_AOac_to_GLac(11405,11405).
map_AOac_to_GLac(11406,11406).
map_AOac_to_GLac(11407,11407).
map_AOac_to_GLac(11408,11408).
map_AOac_to_GLac(11409,11409).
map_AOac_to_GLac(11410,11410).
map_AOac_to_GLac(11411,11411).
map_AOac_to_GLac(11412,11412).
map_AOac_to_GLac(11413,11413).
map_AOac_to_GLac(11414,11414).
map_AOac_to_GLac(11500,11500).
map_AOac_to_GLac(11600,11600).
map_AOac_to_GLac(11700,11700).
map_AOac_to_GLac(11701,11800).
map_AOac_to_GLac(12000,12000).
map_AOac_to_GLac(12100,12100).
map_AOac_to_GLac(12200,12200).
map_AOac_to_GLac(12201,12200).
map_AOac_to_GLac(12401,12401).
map_AOac_to_GLac(12402,12402).
map_AOac_to_GLac(12403,12403).
map_AOac_to_GLac(12600,12600).
map_AOac_to_GLac(12601,12601).
map_AOac_to_GLac(12602,12602).
map_AOac_to_GLac(12801,12801).
map_AOac_to_GLac(2101,2101).
map_AOac_to_GLac(21102,21102).
map_AOac_to_GLac(21103,21103).
map_AOac_to_GLac(21104,21104).
map_AOac_to_GLac(21105,21105).
map_AOac_to_GLac(21106,21106).
map_AOac_to_GLac(21107,21107).
map_AOac_to_GLac(21108,21108).
map_AOac_to_GLac(21201,21201). /* office wages payable */
map_AOac_to_GLac(21202,21202).
map_AOac_to_GLac(21300,21300).
map_AOac_to_GLac(21400,21400).
map_AOac_to_GLac(21502,21502).
map_AOac_to_GLac(21503,21503).
map_AOac_to_GLac(22100,22100).
map_AOac_to_GLac(31100,31100).
map_AOac_to_GLac(31101,31200).
map_AOac_to_GLac(31102,31101). /* Issued Cap */
map_AOac_to_GLac(31103,31101).
map_AOac_to_GLac(31202,31800).

Trading_database - 16
map AOac to GLac(31203,31800).

/* mappings from assets to expenses NOT USED */
map AOac to GLac(11501,33406).
map AOac to GLac(11502,33303).
map AOac to GLac(11601,33306).
map AOac to GLac(11602,33302).
map AOac to GLac(11603,33302).

map AOac to GLac(11701,33401).
map AOac to GLac(11702,33406).
map AOac to GLac(11900,33304).
map AOac to GLac(11901,33402).
map AOac to GLac(11902,33402).
map AOac to GLac(11903,33402).

map AOac to GLac(22101,33501).

map AC_concept to GLac('Accrued Wages',21200).
map AC_concept to GLac('Sales Revenue',32100).
map AC_concept to GLac('Discount Expense',32200).
map AC_concept to GLac('Sales Return',32300).

/*
  map AC_concept to GLac('Bad Debt Expense',32400).
*/
map AC_concept to GLac('Doubtful Debts Expense',32400).
map AC_concept to GLac('Cost of Sales',33100).
map AC_concept to GLac('Retained Earnings',31800).
map AC_concept to GLac('Profit and Loss',31900).
map AC_concept to GLac('Prior Period Gearing Gain',31400).
map AC_concept to GLac('Gearing Gain',31401).
map AC_concept to GLac('Prior Period CCA Gain above General Prices',31500).

/*
  map AC_concept to GLac('CCA Gain above General Prices',31501).
*/
map AC_concept to GLac('Real Holding Gain',31501).
map AC_concept to GLac('CCA Interindex Gain',31600).
map AC_concept to GLac('Prior Period Exchange Rate Fluctuation',31700).
map AC_concept to GLac('Exchange Rate Fluctuation',31701).
map AC_concept to GLac('Opening Balance Fudge',31300).
map AC_concept to GLac('Provision for Doubtful Debts',11300).
map AC_concept to GLac('Non-operating Income',32500).
map AC_concept to GLac('Dividend Received',32600).
map AC_concept to GLac('Abnormal Expense - depreciation adjustment',32700).
map AC_concept to GLac('Discount Revenue',33200).
map AC_concept to GLac('Stocktake Gain/Loss',33405).
map AC_concept to GLac('Donation Expense',33408).
map AC_concept to GLac('Share Premium',31102).
map AC_concept to GLac('Share Discount',31103).
map AC_concept to GLac('Standard Cost Revaluation', 'not defined in GL').
map AC_concept to GLac('Lease Expense',2706). /* cf. 2410 lease obligation */
map AC_concept to GLac('Labour Clearing',3000).
map AC_concept to GLac('Materials Price Variance',3001).
map_AC_concept_to_GLac('Labour Rate Variance',3002).

/* Table 4.3 map_etype_to_eproc(Event_type,event_handler). */

/* Mapping from Event type to Event Procedure in Model1. */

/* special mappings */
map_etype_to_eproc('Credit Sale',credit_sale).
map_etype_to_eproc('Cash Sale',credit_sale).
map_etype_to_eproc('Sale of Depreciable Asset',sale_depreciable_asset).
map_etype_to_eproc('Open Asset and Obligation Accounts',open_acs).
map_etype_to_eproc('Open Amortizable Asset Account',open_amortizable_asset_acs).
map_etype_to_eproc('Cash Receipt',cash_receipt).
map_etype_to_eproc('Cash Payment',cash_payment).
map_etype_to_eproc('Pay Supplier',cash_payment).
map_etype_to_eproc('Wages Payment',simple).
map_etype_to_eproc('Record Direct Labour',record_direct_labour).
map_etype_to_eproc('Record Factory Wages',record_factory_wages).
map_etype_to_eproc('Interdepartmental Goods Transfer',interdept_transfer).
map_etype_to_eproc('Transfer to Finished Goods',transfer_to_FG).
map_etype_to_eproc('Production Order Completion',production_order_completion).
map_etype_to_eproc('Spillage',no_journal_entry_required).
map_etype_to_eproc('Return to RM Store',no_journal_entry_required).
map_etype_to_eproc('Recognize Idle Time',no_journal_entry_required).
map_etype_to_eproc('Issue Shares in Exchange for Factory',issue_shares).
map_etype_to_eproc('Issue Shares',issue_shares).
map_etype_to_eproc('Pay Interest on Debentures',expense_immediately).
map_etype_to_eproc('Pay Advertising Costs',expense_immediately).
map_etype_to_eproc('Advertise ment Run',expense_immediately).
map_etype_to_eproc('Declare Dividend',simple).
map_etype_to_eproc('Receive Dividend',receive_dividend).
map_etype_to_eproc('Lease Tangible Asset',lease_tangible_asset).
map_etype_to_eproc('Lease Payment',lease_payment).
map_etype_to_eproc('Interest Payment',interest_payment).
map_etype_to_eproc('Purchase Return for Credit',purchase_return_for_credit).
map_etype_to_eproc('Sales Return for Credit',sales_return_for_credit).
map_etype_to_eproc('Asset Consumption No Benefit',asset_consumption_no_benefit).
map_etype_to_eproc('Donation',donation).
map_etype_to_eproc('Sell Goods Under Warranty',no_journal_entry_required).

/* simple */
map_etype_to_eproc('Cash Purchase',simple).
map_etype_to_eproc('Credit Purchase',simple).
map_etype_to_eproc('Receive Goods',simple).
map_etype_to_eproc('Receive Supplier Invoice',simple).
map_etype_to_eproc('Pay Employee',simple).
map_etype_to_eproc('Issue Debentures',simple).
map_etype_to_eproc('Pay Factory Rent',simple).
map_etype_to_eproc('Advertise ment Run',simple).

Trading_database - 18
map_etype_to_eproc('Petty Cash Issue',simple).
map_etype_to_eproc('Store Supplies Consumption',simple).

/* no_journal_entry_required */
map_etype_to_eproc('Sales Order',no_journal_entry_required).
map_etype_to_eproc('Purchase Order',no_journal_entry_required).
map_etype_to_eproc('Production Order',no_journal_entry_required).
map_etype_to_eproc('Record Machine Hours',no_journal_entry_required).

/*---------------------------------------------------------------*/
Table 4.4 unit_standard_cost(Rno,Date_set,Standard_cost). */
unit_standard_cost(1020,860301,40.0). /* poly pellets */
unit_standard_cost(1021,860301,0.22). /* cartons */
unit_standard_cost(1052,860301,30.00). /* carton metal caps */
unit_standard_cost(1502,860301,5.00). /* moulding labour */
unit_standard_cost(1503,860301,8.00). /* m/c #1; str line depr = $833 p.m. = $5.2/hr */
unit_standard_cost(1505,860301,4.00). /* packing labour */
unit_standard_cost(1060,860301,10.00). /* pallet, charged out at $10 */

/*---------------------------------------------------------------*/
Table 4.5 departmental_OH_rate(Department,Date_set,Rate,Basis_of_variation). */
departmental_OH_rate('Moulding',860301,8.0,'dlh').
departmental_OH_rate('Packing',860301,2.0,'dlh').

/*---------------------------------------------------------------*/
Table 4.6 department_OH_flexible_budget(Dept,Fixed_OH,Var_OH_coefficient,Basis_of_variation). */
department_OH_flexible_budget('Moulding',1000,4,'standard dlh').
department_OH_flexible_budget('Packing',200,1,'standard dlh').

/*---------------------------------------------------------------*/
Table 4.7 price_index(Series,Date_set,Index_value). */
price_index(g,760101,40).
price_index(g,860101,100).
price_index(g,860201,101).
price_index(g,860301,102).
price_index(g,860401,103).
price_index(g,860501,104).
price_index(g,880101,124).
price_index(g,760101,140).
price_index(g,900101,100).
price_index(g,940101,140).
price_index(sp,760101,50).
price_index(sp,900101,100).
price_index(sp,940101,140).
price_index(sw,760101,30).
848  price_index(sw,900101,100).
849  price_index(sw,940101,150).
850  price_index(s6,800101,100).
851  price_index(s6,900101,200).
852  price_index(s7,800101,100).
853  price_index(s7,900101,200).
854  price_index(s8,800101,100).
855  price_index(s8,900101,200).
856  price_index(s9,800101,100).
857  price_index(s9,900101,200).
858  price_index(g,900101,100).
859  price_index(g,940101,120).
860  price_index(sw,760101,30).
861  price_index(sw,860101,100).
862  price_index(sw,880101,120).

863  /*-----------------------------------------------*/
864  /* Table 4.8 resource_price_level_series_mapping(Rtype,Series). */
865  /* need a line for every non-monetary leaf Rtype */
866  resource_price_level_series_mapping('Cash',g).
867  resource_price_level_series_mapping('Cash at Bank, Current A/C',g).
868  resource_price_level_series_mapping('Cash at Bank, Interest Earning',g).
869  resource_price_level_series_mapping('Trade Debt',g).
870  resource_price_level_series_mapping('Other Debt',g).
871  resource_price_level_series_mapping('Inventory',si).
872  resource_price_level_series_mapping('Raw Materials Inventory (identified)',si).
873  resource_price_level_series_mapping('WIP Inventory (identified)',si).
874  resource_price_level_series_mapping('Finished Goods Inventory (identified)',si).
875  resource_price_level_series_mapping('Raw Materials Inventory',si).
876  resource_price_level_series_mapping('WIP Inventory',si).
877  resource_price_level_series_mapping('Finished Goods Inventory',si).
878  resource_price_level_series_mapping('Stores Inventory',si).
879  resource_price_level_series_mapping('Store Supplies Services',si).
880  resource_price_level_series_mapping('Office Supplies Inventory',si).
881  resource_price_level_series_mapping('Shipping Equipment Inventory',g).
882  resource_price_level_series_mapping('Farm Animals',g).
883  resource_price_level_series_mapping('Land',g).
884  resource_price_level_series_mapping('Building',sp).
885  resource_price_level_series_mapping('Plant',sp).
886  resource_price_level_series_mapping('Vehicle',sp).
887  resource_price_level_series_mapping('Office Equipment',sp).
888  resource_price_level_series_mapping('Store Equipment',sp).
889  resource_price_level_series_mapping('Debenture',g).
890  resource_price_level_series_mapping('Lease',g).
891  resource_price_level_series_mapping('Loan',g).
892  resource_price_level_series_mapping('Share (stock)',g).
893  resource_price_level_series_mapping('Proprietorship Unit',g).

Trading_database - 20
resource_price_level_series_mapping('Labour Services',sw).
resource_price_level_series_mapping('Occupancy Rights',g).
resource_price_level_series_mapping('Warranty',g).
resource_price_level_series_mapping('Goodwill',g).
resource_price_level_series_mapping('Informal Customer',g).
resource_price_level_series_mapping('Customer Goodwill',g).
resource_price_level_series_mapping('Delivery Services',g).
resource_price_level_series_mapping('Insurance Services',g).
resource_price_level_series_mapping('Banking Services',sw).
resource_price_level_series_mapping('Stores Supplies Services',g).

/** -----------------------------------------------*/
/* Table 4.9 bad_debts_allowance(Over x months,Percent uncollectable).*/
bad_debts_allowance(3,50).
bad_debts_allowance(2,25).
bad_debts_allowance(1,10).
bad_debts_allowance(0,3).

/** -----------------------------------------------*/
/* Table 4.10 foreign_exchange_rate(Currency,Date,Buyer of home currency_pays,Seller of home_currency_pays).*/
foreign_exchange_rate('Yen',860806,126.08,124.14).
foreign_exchange_rate('Yen',870806,106.08,104.14).
foreign_exchange_rate('$US',860806,0.6565,0.6915).
foreign_exchange_rate('$US',870806,0.6965,0.6915).
foreign_exchange_rate('Pound',860806,0.4000,0.3950).
foreign_exchange_rate('Pound',870806,0.4431,0.4381).
foreign_exchange_rate('Dmark',860806,1.2198,1.1959).
foreign_exchange_rate('Dmark',870806,1.3198,1.2959).

/** -----------------------------------------------*/
/* Table 4.11 nominal_discount_rate(Security_class,Date,Annual_rate).*/
nominal_discount_rate('Long Term Debt',870101,0.14).
nominal_discount_rate('Short Term Debt',870101,0.13).
nominal_discount_rate('Equity',870101,0.16).

done(tables4).
Index to "Tables" in the Trading Database

agent(Agent_id,Agent_name,Address) ........................................... 13
allocation_of_OH(From_GLac,To_GLac,Proportion) ......................... 6
ao_account(ACno,Description,Class,Rno,Agent_id,Controller) ............ 7
ao_hierarchy(Pl,Pname,Cseq,Cname) ........................................... 8
bad_debts_allowance(Over_x_months,Percent_uncollectable) ............... 21
chart_of_accounts(GLacno,description,class) ................................ 2
composite_resource(Rno,Component_no,Date_defined,Component_Rno) .... 13
department_GL_variance_account(Department,Variance_ac_description) ... 7
department_OH_flexible_budget(Dept,Fixed_OH,Var_OH_coefficient) ....... 19
departmental_OH_rate(Department,Date_set,Rate,Basis_of_variation) .... 19
eh(Eno,Date,Etype,Doc_type,Doc_no,Agent_id,Narrative) ................. 15
expected_value(Rno,Expected_pattern_of_benefits,Expected_life) ...... 14
expense_ac_for_growing_liability(GL_ACno,Expense_ACno) ............. 6
foreign_exchange_rate(Currency,Date,Buyer_of_home_currency_pay) ... 21
gl_amortization(GL_ACno,Expense_ACno) ................................... 6
gl_depreciation(Asset_ACno,Accum_depr_ACno,Depr_exp_ACno) .......... 6
glAC_hierarchy(P_seq,P_name,C_seq,C_name) ................................ 4
growing_resource(Rno,Expected_life_in_months,Annual_growth_rate) ... 14
jed(Eno,Line_no,JE_ACno,Amount) */ */ generated by Interpreter .... 6
jeh(Eno,Date,Eno,Narration) ................................................... 6
map_AC_concept_to_GLac(Accounting_concept,GLac) ....................... 17
map_AOac_to_GLac(AOac,GLac) ................................................ 15
map_etype_to_event(Event_type,event_handler) .............................. 18
nominal_discount_rate(Security_class,Date,Annual_rate) .................... 21
price_index(Series,Date_set,Index_value) .................................... 19
promised_resource(Promised_Rno,Component_no,Date_defined,Component) 14
rc(Eno,Line_no,AOacno,Qty,Value) ........................................... 15
resource(Rno,Description,Unit_of_measure,RTtype) ....................... 10
resource_ev_hierarchy(PI,Pname,Cseq,Cname) ................................ 12
resource_hierarchy(Pl,Pname,Cseq,Cname) .................................. 11
resource_price_level_series_mapping(Rtype,Series) ....................... 20
share_stock(Rno,Par_value,Date_incorporated) ................................ 14
tied_asset_obligation(ACno1,ACno2) ......................................... 13
unit_standard_cost(Rno,Date_set,Standard_cost) ............................ 19
wip_GLac_class(GL_ACno,WIP_class) ........................................... 6
APPENDIX 6: MANUFACTURING DATABASE

This Appendix contains 873 lines (20 pages) of Prolog facts (equivalent to relational database tables) for the Interpreter. Tables have been organized into logically-related groups as follows:

<table>
<thead>
<tr>
<th>Tables</th>
<th>pages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formula Accounting</strong></td>
<td>2-7</td>
</tr>
<tr>
<td>This section contains tables for the Formula Accounting system, as defined in Chapter 8, Figure 8.8 except for those created by the Interpreter. All these tables are headed Table 1.xx.</td>
<td></td>
</tr>
<tr>
<td><strong>Resources and Exchange Events Model</strong></td>
<td>7-14</td>
</tr>
<tr>
<td>This section contains tables for the REE Accounting model, as defined in Chapter 9, Figure 9.9. All these tables are headed Table 2.xx or Table 3.xx.</td>
<td></td>
</tr>
<tr>
<td>This section contains tables for the Manufacturing Exchange Events from Chapter 9, Table 9.2. eh(...) and rc(...) predicates have been interleaved to match Table 9.2.</td>
<td>14-15</td>
</tr>
<tr>
<td><strong>Interpreter</strong></td>
<td>15-21</td>
</tr>
<tr>
<td>This section contains tables that are not part of either the FA system or for the REE model, but which are required by the Interpreter, e.g., tables to map from one model to the other, from accounting concepts to GL accounts, and to invoke the correct procedure to process an Exchange Event. All these tables are headed Table 4.xx.</td>
<td></td>
</tr>
</tbody>
</table>

Here is an index to all tables in this Appendix on page 22.
Table 1.1: chart_of_accounts(GLAcno,description_class). */

chart_of_accounts(2000, 'Cash at Bank A', 'Cash at Bank').
chart_of_accounts(2010, 'Customer C', 'Accounts Receivable - Gross').
chart_of_accounts(2012, 'Provision for Doubtful Debts', 'Provision for Doubtful Debts').
chart_of_accounts(2021, 'Polyethylene Pellets', 'Raw Materials Inventory').
chart_of_accounts(2022, 'Empty Cartons', 'Raw Materials Inventory').

chart_of_accounts(2030, 'Materials in Progress, Job 122, Step 1', 'WIP - Job 122, Step 1').
chart_of_accounts(2031, 'Labour in Progress, Job 122, Step 1', 'WIP - Job 122, Step 1').
chart_of_accounts(2035, 'Materials in Progress, Job 122, Step 2', 'WIP - Job 122, Step 2').
chart_of_accounts(2036, 'Labour in Progress, Job 122, Step 2', 'WIP - Job 122, Step 2').

chart_of_accounts(2040, 'Materials in Progress, Job 123, Step 1', 'WIP - Job 123, Step 1').
chart_of_accounts(2041, 'Labour in Progress, Job 123, Step 1', 'WIP - Job 123, Step 1').
chart_of_accounts(2045, 'Materials in Progress, Job 123, Step 2', 'WIP - Job 123, Step 2').
chart_of_accounts(2046, 'Labour in Progress, Job 123, Step 2', 'WIP - Job 123, Step 2').

chart_of_accounts(2071, 'Carton 1 Litre Bottles', 'Finished Goods Inventory').
chart_of_accounts(2072, 'Carton 2 Litre Bottles', 'Finished Goods Inventory').
chart_of_accounts(2073, 'Carton Metal Caps', 'Finished Goods Inventory').
chart_of_accounts(2090, 'Prepaid Rent', 'Prepaid Rent').

chart_of_accounts(2100, 'Blow Moulding Machine 1', 'Plant - Gross').
chart_of_accounts(2101, 'Blow Moulding Machine 2', 'Plant - Gross').
chart_of_accounts(2102, 'Truck, Reg No. ABC123', 'Vehicles - Gross').
chart_of_accounts(2103, 'Car, Reg No. ABD456', 'Vehicles - Gross').
chart_of_accounts(2104, 'Car, Reg No. ABD457', 'Vehicles - Gross').
chart_of_accounts(2110, 'Accumulated Depreciation Machine 1', 'Accumulated Depreciation - Plant').
chart_of_accounts(2111, 'Accumulated Depreciation Machine 2', 'Accumulated Depreciation - Plant').
chart_of_accounts(2112, 'Accumulated Depreciation Truck ABC123', 'Accumulated Depreciation - Vehicles').
chart_of_accounts(2113, 'Accumulated Depreciation Car ABD456', 'Accumulated Depreciation - Vehicles').
chart_of_accounts(2114, 'Accumulated Depreciation Car ABD457', 'Accumulated Depreciation - Vehicles').
chart_of_accounts(2120, 'Factory Building, 25 Keys Rd', 'Factory Building').
chart_of_accounts(2200, 'Goodwill', 'Goodwill').

chart_of_accounts(2300, 'Supplier C', 'Accounts Payable').
chart_of_accounts(2301, 'Supplier E', 'Accounts Payable').
chart_of_accounts(2302, 'Supplier F', 'Accounts Payable').
chart_of_accounts(2303, 'Supplier X', 'Accounts Payable').
chart_of_accounts(2310, 'Employee Taxes Payable', 'Tax Payable').
chart_of_accounts(2311, 'Factory Wages Payable', 'Wages Payable').
chart_of_accounts(2350, 'Dividend Payable - C', 'Dividend Payable').
chart_of_accounts(2351, 'Dividend Payable - P', 'Dividend Payable').
chart_of_accounts(2401,'Debenture Holder G','Debenture').
chart_of_accounts(2402,'Debenture Holder H','Debenture').
chart_of_accounts(2410,'Leasor - D, Car ABD457','Lease').
chart_of_accounts(2500,'Issued Capital C','Issued Capital').
chart_of_accounts(2501,'Issued Capital P','Issued Capital').
chart_of_accounts(2600,'Retained Earnings','Retained Earnings').
chart_of_accounts(2601,'Asset Revaluation','Asset Revaluation').
chart_of_accounts(2602,'Share Premium Reserve','Share Premium Reserve').
chart_of_accounts(2603,'Gearing Gain','Gearing Gain').
chart_of_accounts(2604,'CCA Gain above General Prices','Retained Earnings').
chart_of_accounts(2605,'CCA Standard Cost Interindex Gain','Revenue').
chart_of_accounts(2606,'Opening Balances Fudge','Opening Balances Fudge').
chart_of_accounts(2700,'Sales Revenue','Revenue').
chart_of_accounts(2701,'Discount Revenue','Revenue').
chart_of_accounts(2702,'Rent Expense','Rent Expense').
chart_of_accounts(2703,'Interest Expense','Interest Expense').
chart_of_accounts(2704,'Advertising Expense','Advertising Expense').
chart_of_accounts(2705,'Discount Expense','Discount Expense').
chart_of_accounts(2706,'Lease Expense','Lease Expense').
chart_of_accounts(2707,'Office Wages Expense','Wages Expense').
chart_of_accounts(2710,'Profit and Loss','Profit and Loss').
chart_of_accounts(2711,'Standard Cost of Sales','Cost of Sales').
chart_of_accounts(2712,'Non-operating Income','Non-operating Income').
chart_of_accounts(2713,'Dividend Received','Non-operating Income').
chart_of_accounts(2790,'Depreciation Expense - Despatch','Despatch Expense').
chart_of_accounts(2801,'Depreciation Expense - Marketing','Marketing Expense').
chart_of_accounts(2802,'Production OH - General','Production OH').
chart_of_accounts(2803,'Production OH - Contract','Production OH').
chart_of_accounts(2810,'Overhead Incurred - Moulding','Production OH').
chart_of_accounts(2811,'Overhead Applied - Moulding','Production OH').
chart_of_accounts(2813,'Materials Usage Variance - Moulding','Standard Cost Variance').
chart_of_accounts(2814,'Labour Usage Variance - Moulding','Standard Cost Variance').
chart_of_accounts(2815,'OH Spending Variance - Moulding','Standard Cost Variance').
chart_of_accounts(2816,'OH Labour Efficiency Variance - Moulding','Standard Cost Variance').
chart_of_accounts(2817,'OH Volume Variance - Moulding','Standard Cost Variance').
chart_of_accounts(2820,'Overhead Incurred - Packing','Production OH').
chart_of_accounts(2821,'Overhead Applied - Packing','Production OH').
chart_of_accounts(2823,'Materials Usage Variance - Packing','Standard Cost Variance').
chart_of_accounts(2824,'Labour Usage Variance - Packing','Standard Cost Variance').
chart_of_accounts(2825,'OH Spending Variance - Packing','Standard Cost Variance').
chart_of_accounts(2826,'OH Labour Efficiency Variance - Packing','Standard Cost Variance').
chart_of_accounts(2827,'OH Volume Variance - Packing','Standard Cost Variance').
chart_of_accounts(2830,'Office Share of Rent','Rent Expense').
chart_of_accounts(2900,'Amortization of Goodwill','Amortization Expense').
chart_of_accounts(3000,’Labour Clearing’,’Labour Clearing’).
chart_of_accounts(3001,’Materials Price Variance’,’Standard Cost Variance’).
chart_of_accounts(3002,’Labour Rate Variance’,’Standard Cost Variance’).

/*
 Table 1.2: glAC_hierarchy(P_seq,P_name,C_seq,C_name).
*/

glAC_hierarchy(0,’Account’,1,’Asset’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(0,’Account’,3,’Owners Equity’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(0,’Account’,2,’Liability’).

glAC_hierarchy(1,’Asset’,1,’Current Asset’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(1,’Asset’,2,’Fixed Asset’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(1,’Asset’,3,’Other Asset’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(1,’Liability’,1,’Current Liability’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(1,’Liability’,2,’Long Term Liability’).

glAC_hierarchy(1,’Owners Equity’,1,’Issued Capital’).

glAC_hierarchy(1,’Owners Equity’,2,’Retained Earnings’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(1,’Owners Equity’,3,’Share Premium Reserve’).

glAC_hierarchy(1,’Owners Equity’,4,’Asset Revaluation’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(1,’Owners Equity’,5,’Gearing Gain’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(1,’Owners Equity’,6,’Opening Balances Fudge’).

glAC_hierarchy(1,’ Owners Equity’,7,’Operating Income’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(1,’ Owners Equity’,8,’Non-operating Income’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(1,’ Owners Equity’,7,’Profit and Loss’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Current Asset’,1,’Cash at Bank’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Current Asset’,2,’Accounts Receivable’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Current Asset’,3,’Inventory’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Current Asset’,4,’Prepaid Rent’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Fixed Asset’,1,’Factory Building’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Fixed Asset’,2,’Plant’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Fixed Asset’,3,’Vehicles’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Other Asset’,1,’Goodwill’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Long Term Liability’,1,’Debenture’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Long Term Liability’,2,’Lease’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Operating Income’,1,’Gross Profit’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(2,’Operating Income’,2,’Expense’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(3,’Gross Profit’,1,’Revenue’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(3,’Gross Profit’,2,’Cost of Sales’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(3,’Accounts Receivable’,1,’Accounts Receivable - Gross’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(3,’Accounts Receivable’,2,’Provision for Doubtful Debts’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(3,’Inventory’,1,’Raw Materials Inventory’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(3,’Inventory’,2,’WIP Inventory’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(3,’Inventory’,3,’Finished Goods Inventory’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(3,’Plant’,1,’Plant - Gross’).

/* deliberately out of seq to test reporting */
glAC_hierarchy(3,’Plant’,2,’Accumulated Depreciation - Plant’).
glAC_hierarchy(3,'Vehicles',1,'Vehicles - Gross').
glAC_hierarchy(3,'Vehicles',2,'Accumulated Depreciation - Vehicles').

*/ The 3 for P_level is deliberately wrong; logically it should be 2. */
*/ See result in Col_from_right of HB.PL (Hierarchy balances) */
glAC_hierarchy(3,'Current Liability',1,'Accounts Payable').
glAC_hierarchy(3,'Current Liability',2,'Tax Payable').
glAC_hierarchy(3,'Current Liability',3,'Dividend Payable').
glAC_hierarchy(3,'Current Liability',4,'Wages Payable').
glAC_hierarchy(3,'Current Liability',5,'Labour Clearing').

glAC_hierarchy(3,'Expense',1,'Rent Expense').
glAC_hierarchy(3,'Expense',2,'Interest Expense').
glAC_hierarchy(3,'Expense',3,'Advertising Expense').
glAC_hierarchy(3,'Expense',4,'Discount Expense').
glAC_hierarchy(3,'Expense',5,'Amortization Expense').
glAC_hierarchy(3,'Expense',6,'Lease Expense').
glAC_hierarchy(3,'Expense',7,'Standard Cost Variance').
glAC_hierarchy(3,'Expense',8,'Production OH').
glAC_hierarchy(3,'Expense',9,'Marketing Expense').
glAC_hierarchy(3,'Expense',10,'Despatch Expense').
glAC_hierarchy(3,'Expense',11,'Wages Expense').

glAC_hierarchy(4,'WIP Inventory',1,'WIP - Job 122').
glAC_hierarchy(4,'WIP Inventory',2,'WIP - Job 123').

glAC_hierarchy(5,'WIP - Job 122',1,'WIP - Job 122, Step 1').
glAC_hierarchy(5,'WIP - Job 122',2,'WIP - Job 122, Step 2').
glAC_hierarchy(5,'WIP - Job 123',1,'WIP - Job 123, Step 1').
glAC_hierarchy(5,'WIP - Job 123',2,'WIP - Job 123, Step 2').

/* Table 1.3 jeh(JEno,Date,Eno,Narration). */
*/ to be created by Interpreter */

/* Table 1.4 jed(JEno,Line_no,JE_ACno,Amount). */
*/ to be created by Interpreter */

/* Table 1.5 gl depreciation(Asset_ACno,Accum_Depr_ACno,Depr_Exp_ACno). */

gl_depreciation(2100,2110,2810).
gl_depreciation(2101,2111,2810).
gl_depreciation(2102,2112,2800).
gl_depreciation(2103,2113,2801).
gl_depreciation(2104,2114,2802).

/* Table 1.7 gl amortization(GL_ACno,Expense_ACno). */

gl_amortization(2090,2802).
gl_amortization(2200,2900).

/* Table 1.8 expense_ac_for_growing_liability(GL_ACno,Expense_ACno). */
Table 1.9 wip_GLac_class(GL_ACno,WIP_class). (Prime costs only).

Table 1.10 allocation_of_OH(From_GLAc,To_GLAc,Proportion).

Table 1.11 department_GL_variance_account(Department,Variance_ac_description,GL_Acno).
Table 1.12

<table>
<thead>
<tr>
<th>Production Step GLacs</th>
<th>(Job_no, Step_no, M_Glac, L_Glac, OH_Incurred_GLac, OH_Applied_GLac)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLacs 122,1,2030,2031,2810,2811</td>
<td>moulding</td>
</tr>
<tr>
<td>GLacs 122,2,2035,2036,2820,2821</td>
<td>packing</td>
</tr>
<tr>
<td>GLacs 123,1,2040,2041,2810,2811</td>
<td>moulding</td>
</tr>
<tr>
<td>GLacs 123,2,2045,2046,2820,2821</td>
<td>packing</td>
</tr>
</tbody>
</table>

Table 2.1 ao_account(AcNo, Description, Class, Rno, Agent_id, Controller)

<table>
<thead>
<tr>
<th>AcNo</th>
<th>Description</th>
<th>Class</th>
<th>Rno</th>
<th>Agent_id</th>
<th>Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cash at Bank A, A/c# 1234</td>
<td>Cash at Bank</td>
<td>1002</td>
<td>A</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>2</td>
<td>Cash at Bank A, A/c# 4567</td>
<td>Cash at Bank</td>
<td>1002</td>
<td>B</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>11</td>
<td>Customer C - debt</td>
<td>Accounts Receivable</td>
<td>1012</td>
<td>C</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>12</td>
<td>Customer D - debt</td>
<td>Accounts Receivable</td>
<td>1012</td>
<td>D</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>21</td>
<td>Customer C - Sales Order</td>
<td>Sales Order</td>
<td>1012</td>
<td>C</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>31</td>
<td>Polyethelene Pellets</td>
<td>Raw Materials Inventory</td>
<td>1020</td>
<td>self</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>32</td>
<td>Polyethelene Pellets</td>
<td>Raw Materials Inventory</td>
<td>1021</td>
<td>self</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>51</td>
<td>Carton 1 Litre Bottles</td>
<td>Finished Goods Inventory</td>
<td>1050</td>
<td>self</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>52</td>
<td>Carton 2 Lire Bottles</td>
<td>Finished Goods Inventory</td>
<td>1051</td>
<td>self</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>53</td>
<td>Carton Metal Caps</td>
<td>Finished Goods Inventory</td>
<td>1052</td>
<td>self</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>61</td>
<td>Polyethylene Pellets - on Order</td>
<td>Goods on Order</td>
<td>1020</td>
<td>E</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>62</td>
<td>Carton Metal Caps - on Order</td>
<td>Goods on Order</td>
<td>1052</td>
<td>F</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>70</td>
<td>Factory Usage Rights (Landlord P)</td>
<td>Occupancy Rights</td>
<td>1501</td>
<td>self</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>72</td>
<td>Office Usage Rights (Landlord P)</td>
<td>Occupancy Rights</td>
<td>1504</td>
<td>self</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>80</td>
<td>Advertising - Campaign 7</td>
<td>Advertising</td>
<td>1500</td>
<td>self</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>81</td>
<td>Office Labour Services</td>
<td>Labour Services</td>
<td>1506</td>
<td>M</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>101</td>
<td>Blow Moulding Machine 1</td>
<td>Plant</td>
<td>1100</td>
<td>self</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>102</td>
<td>Blow Moulding Machine 2</td>
<td>Plant</td>
<td>1101</td>
<td>self</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>103</td>
<td>Truck, Reg No. ABC123</td>
<td>Vehicles</td>
<td>1102</td>
<td>self</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>104</td>
<td>Car, Reg No. ABD456</td>
<td>Vehicles</td>
<td>1103</td>
<td>self</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>105</td>
<td>Factory, 25 Keys Road,...</td>
<td>Land and Buildings</td>
<td>1150</td>
<td>self</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>106</td>
<td>Car, Reg No. ABD457</td>
<td>Vehicles</td>
<td>1104</td>
<td>self</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>120</td>
<td>Goodwill</td>
<td>Goodwill</td>
<td>1200</td>
<td>self</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>201</td>
<td>Supplier C (debt to)</td>
<td>Accounts Payable</td>
<td>1012</td>
<td>C</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>202</td>
<td>Supplier E (debt to)</td>
<td>Accounts Payable</td>
<td>1012</td>
<td>E</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>203</td>
<td>Supplier F (debt to)</td>
<td>Accounts Payable</td>
<td>1012</td>
<td>F</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>204</td>
<td>Supplier X (debt to)</td>
<td>Accounts Payable</td>
<td>1012</td>
<td>X</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>211</td>
<td>Supplier C - Purchase Order</td>
<td>Purchase Order</td>
<td>1012</td>
<td>C</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>212</td>
<td>Supplier E - Purchase Order</td>
<td>Purchase Order</td>
<td>1012</td>
<td>E</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>213</td>
<td>Supplier F - Purchase Order</td>
<td>Purchase Order</td>
<td>1012</td>
<td>F</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>214</td>
<td>Supplier X - Purchase Order</td>
<td>Purchase Order</td>
<td>1012</td>
<td>X</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>220</td>
<td>Dividend Payable - C</td>
<td>Dividend Payable</td>
<td>1010</td>
<td>C</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>221</td>
<td>Dividend Payable - P</td>
<td>Dividend Payable</td>
<td>1010</td>
<td>P</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>231</td>
<td>Carton 2L Bottles on Order by C</td>
<td>Goods</td>
<td>1051</td>
<td>C</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>250</td>
<td>Wages Payable - M</td>
<td>Wages Payable</td>
<td>1010</td>
<td>M</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>251</td>
<td>Wages Payable - N</td>
<td>Wages Payable</td>
<td>1010</td>
<td>N</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>252</td>
<td>Wages Payable - Q</td>
<td>Wages Payable</td>
<td>1010</td>
<td>Q</td>
<td>Non-Manufacturing</td>
</tr>
<tr>
<td>260</td>
<td>Employee Taxes Payable</td>
<td>Tax Payable</td>
<td>1010</td>
<td>T</td>
<td>Non-Manufacturing</td>
</tr>
</tbody>
</table>

Manuf_database - 7
ao_account(262,'Interest Payable - G','Interest Payable',1014,'G','Non-Manufacturing').
ao_account(270,'Warranty - D','Warranty',1701,'D','Non-Manufacturing').
ao_account(281,'Debenture Holder G','Debentures',1300,'G','Non-Manufacturing').
ao_account(282,'Debenture Holder H','Debentures',1300,'H','Non-Manufacturing').

ao_account(290,'Lease Obligation - Car ABD456','Lease Obligation',1601,'D','Non-Manufacturing').
ao_account(291,'Lease Obligation - Car ABD457','Lease Obligation',1602,'D','Non-Manufacturing').

ao_account(301,'Stock Holder C','Owners Equity',1400,'C','Non-Manufacturing').
ao_account(302,'Stock Holder P','Owners Equity',1400,'P','Non-Manufacturing').

/* WORK IN PROGRESS */

*/ Very simple 2 step jobs: Step 1 is moulding, Step 2 is packing */

ao_account(410,'Unpacked 2L Bottles - Job 122, Step 2','Resources Committed - Job 122, Step 2',1031,self,'Packing').

ao_account(411,'Empty cartons - Job 122, Step 2','Resources Committed - Job 122, Step 2',1021,self,'Packing').

ao_account(412,'Packing Labour - Job 122, Step 2','Resources Committed - Job 122, Step 2',1505,self,'Packing').

ao_account(419,'Carton 2L Bottles - Job 122, Step 2','Planned Production - Job 122, Step 2',1051,self,'Packing').

/* Job 123, Step 1, Moulding Dept */

ao_account(420,'Polyethylene Pellets - Job 123, Step 1','Resources Committed - Job 123, Step 1',1020,self,'Moulding').

ao_account(421,'Moulding Labour - Job 123, Step 1','Resources Committed - Job 123, Step 1',1502,self,'Moulding').

ao_account(422,'Machine 1 Services - Job 123, Step 1','Resources Committed - Job 123, Step 1',1503,self,'Moulding').

ao_account(429,'Unpacked 1L Bottles - Job 123, Step 1','Planned Production - Job 123, Step 1',1030,self,'Moulding').

/* Job 123, Step 2, Packing Dept */

ao_account(430,'Unpacked 1L Bottles - Job 123, Step 2','Resources Committed - Job 123, Step 2',1030,self,'Packing').

ao_account(431,'Empty cartons - Job 123, Step 2','Resources Committed - Job 123, Step 2',1021,self,'Packing').

ao_account(432,'Packing Labour - Job 123, Step 2','Resources Committed - Job 123, Step 2',1505,self,'Packing').

ao_account(439,'Carton 1L Bottles - Job 123, Step 2','Planned Production - Job 123, Step 2',1050,self,'Packing').

/* Moulding Department Obligation to Employees; Wages Payable (in hours); 1502, self, 'Moulding').

/* Packing Department Obligation to Employees; Wages Payable (in hours); 1505, self, 'Packing').

/* Idle Time - Moulding; Idle Time; 1502, self, 'Moulding').

/* Idle Time - Packing; Idle Time; 1505, self, 'Packing').

/* Table 2.2 ao_hierarchy(PI,Pname,Cseq,Cname).*/

/* Table 2.2 ao_hierarchy(PI,Pname,Cseq,Cname).*/
ao_hierarchy(1,'Asset',2,'Conditional Asset').
ao_hierarchy(1,'Obligation',1,'Unconditional Obligation').
ao_hierarchy(1,'Obligation',2,'Conditional Obligation').
ao_hierarchy(1,'Unconditional Asset',1,'Current Asset').
ao_hierarchy(1,'Unconditional Asset',2,'Fixed Asset').
ao_hierarchy(1,'Unconditional Asset',3,'Other Asset').
ao_hierarchy(2,'Unconditional Obligation',1,'Current Liability').
ao_hierarchy(2,'Unconditional Obligation',2,'Long Term Liability').
ao_hierarchy(2,'Unconditional Obligation',3,'Owners Equity').
ao_hierarchy(2,'Conditional Asset',1,'Sales Order').
ao_hierarchy(2,'Conditional Asset',2,'Goods on Order').
ao_hierarchy(2,'Conditional Asset',3,'Planned Production').
ao_hierarchy(2,'Conditional Obligation',1,'Purchase Order').
ao_hierarchy(2,'Conditional Obligation',2,'Goods Promised').
ao_hierarchy(2,'Conditional Obligation',3,'Resources Committed').
ao_hierarchy(2,'Current Asset',1,'Cash at Bank').
ao_hierarchy(2,'Current Asset',2,'Accounts Receivable').
ao_hierarchy(2,'Current Asset',3,'Inventory').
ao_hierarchy(2,'Current Asset',4,'Occupancy Rights').
ao_hierarchy(2,'Current Asset',5,'Advertising').
ao_hierarchy(2,'Current Asset',6,'Idle Time').
ao_hierarchy(2,'Current Asset',7,'Labour Services').
ao_hierarchy(2,'Fixed Asset',1,'Investment').
ao_hierarchy(2,'Fixed Asset',2,'Plant').
ao_hierarchy(2,'Fixed Asset',3,'Vehicles').
ao_hierarchy(2,'Fixed Asset',4,'Land and Buildings').
ao_hierarchy(2,'Other Asset',1,'Goodwill').
ao_hierarchy(2,'Current Liability',1,'Accounts Payable').
ao_hierarchy(2,'Current Liability',2,'Wages Payable').
ao_hierarchy(2,'Current Liability',3,'Tax Payable').
ao_hierarchy(2,'Current Liability',4,'Dividend Payable').
ao_hierarchy(2,'Current Liability',5,'Interest Payable').
ao_hierarchy(2,'Current Liability',6,'Warranty').
ao_hierarchy(2,'Current Liability',7,'Wages Payable (in hours)').
ao_hierarchy(2,'Long Term Liability',1,'Debentures').
ao_hierarchy(2,'Long Term Liability',2,'Lease Obligation').
ao_hierarchy(2,'Planned Production',1,'Planned Production - Job 122').
ao_hierarchy(2,'Planned Production',2,'Planned Production - Job 123').
ao_hierarchy(2,'Resources Committed',1,'Resources Committed - Job 122').
ao_hierarchy(2,'Resources Committed',2,'Resources Committed - Job 123').
ao_hierarchy(3,'Current Asset',1,'Cash at Bank').
ao_hierarchy(3,'Current Asset',2,'Accounts Receivable').
ao_hierarchy(3,'Current Asset',3,'Inventory').
ao_hierarchy(3,'Current Asset',4,'Occupancy Rights').
ao_hierarchy(3,'Current Asset',5,'Advertising').
ao_hierarchy(3,'Current Asset',6,'Idle Time').
ao_hierarchy(3,'Current Asset',7,'Labour Services').
ao_hierarchy(3,'Fixed Asset',1,'Investment').
ao_hierarchy(3,'Fixed Asset',2,'Plant').
ao_hierarchy(3,'Fixed Asset',3,'Vehicles').
ao_hierarchy(3,'Fixed Asset',4,'Land and Buildings').
ao_hierarchy(3,'Other Asset',1,'Goodwill').
ao_hierarchy(3,'Current Liability',1,'Accounts Payable').
ao_hierarchy(3,'Current Liability',2,'Wages Payable').
ao_hierarchy(3,'Current Liability',3,'Tax Payable').
ao_hierarchy(3,'Current Liability',4,'Dividend Payable').
ao_hierarchy(3,'Current Liability',5,'Interest Payable').
ao_hierarchy(3,'Current Liability',6,'Warranty').
ao_hierarchy(3,'Current Liability',7,'Wages Payable (in hours)').
ao_hierarchy(3,'Long Term Liability',1,'Debentures').
ao_hierarchy(3,'Long Term Liability',2,'Lease Obligation').
ao_hierarchy(3,'Planned Production',1,'Planned Production - Job 122').
ao_hierarchy(3,'Planned Production',2,'Planned Production - Job 123').
ao_hierarchy(3,'Resources Committed',1,'Resources Committed - Job 122').
ao_hierarchy(3,'Resources Committed',2,'Resources Committed - Job 123').
ao_hierarchy(4,'Inventory',1,'Raw Materials Inventory').
ao_hierarchy(4,'Inventory',2,'WIP Inventory').

Manuf_database - 9
/* Table 2.3 resource(Rno,Description,Unit_of_measure,Rtype). */

resource(1001,'Cash on Hand','$','Cash').
resource(1002,'Cash at Bank, Current A/C','$','Cash at Bank, Current A/C').
resource(1003,'Cash at Bank, Interest Earning','$','Cash at Bank, Interest Earning').
resource(1010,'Immediate Debt','$','Other Debt').
resource(1011,'7 Day Debt','$','Trade Debt').
resource(1012,'30 Day Debt','$','Trade Debt').
resource(1013,'30 to 90 Day Debt','$','Other Debt').
resource(1014,'90 Day to 1 Year Debt','$','Other Debt').
resource(1020,'Polyethylene Pellets','20 Kg bag','Raw Materials Inventory').
resource(1021,'Empty Carton','Unit','Raw Materials Inventory').
resource(1030,'Unpacked IL Bottles','Unit','WIP Inventory').
resource(1031,'Unpacked 2L Bottles','Unit','WIP Inventory').
resource(1050,'Carton 1 Litre Bottles','Carton','Finished Goods Inventory').
resource(1051,'Carton 2 Litre Bottles','Carton','Finished Goods Inventory').
resource(1052,'Carton Metal Caps','Carton','Finished Goods Inventory').
resource(1060,'Pallet','Pallet','Shipping Equipment Inventory').
resource(1100,'Blow Moulding Machine 1','Machine','Plant').
resource(1101,'Blow Moulding Machine 2','Machine','Plant').
resource(1102,'Truck, Reg No. ABC123','Truck','Vehicle').
resource(1103,'Car, Reg No. ABD456','Car','Vehicle').
resource(1104,'Car, Reg No. ABD457','Car','Vehicle').
resource(1150,'Factory Building, 25 Keys Rd.','Building','Building').
resource(1200,'Goodwill','$','Goodwill').
resource(1300,'7% Debenture, 5 years (in dollars)','Debenture','Debenture').
resource(1301,'8%, 1000 yen Debenture, 4 years','Debenture','Debenture').
resource(1400,'Ordinary Share (stock) in self','Unit','Share (stock)').
resource(1401,'Share (stock) in R','Unit','Share (stock)').
resource(1500,'Advertising','$','Informed Customer').
resource(1501,'Factory Occupancy Rights','Month','Occupancy Rights').
resource(1502,'Factory Labour Services - Moulding','Hour','Labour Services').
resource(1503,'Machine Services - Machine 1','Hour','Machine Services').
resource(1504,'Office Occupancy Rights','Month','Occupancy Rights').
resource(1505,'Factory Labour Services - Packing','Hour','Labour Services').
resource(1506,'Office Labour Services','Hour','Labour Services').
resource(1601,'Car Lease Receipts Car ABD456','Lease','Lease').
resource(1602,'Car Lease Receipts Car ABD457','Lease','Lease').
resource(1701,'Warranty Protection - 2L bottles','Bottle','Warranty').

/*-----------------------------*/
/* Table 2.4A resource_hierarchy(Pl,Pname,Cseq,Cname).*/

resource_hierarchy(0,'Resource',1,'Monetary Resource').
resource_hierarchy(0,'Resource',2,'Non-monetary (N.M.) Resource').

resource_hierarchy(1,'Monetary Resource',1,'Cash Available Immediately').
resource_hierarchy(1,'Monetary Resource',2,'Single Future Flow of Money').
resource_hierarchy(1,'Monetary Resource',3,'Stream of Future Flows of Money').
resource_hierarchy(1,'Non-monetary Resource',1,'Tangible N.M. Resource').

resource_hierarchy(2,'Cash Available Immediately',1,'Cash').
resource_hierarchy(2,'Cash Available Immediately',2,'Cash at Bank, Current A/C').
resource_hierarchy(2,'Cash Available Immediately',3,'Cash at Bank, Interest Earning').
resource_hierarchy(2,'Single Future Flow of Money',1,'Trade Debt').
resource_hierarchy(2,'Single Future Flow of Money',2,'Other Debt').
resource_hierarchy(2,'Stream of Future Flows of Money',1,'Debenture').
resource_hierarchy(2,'Stream of Future Flows of Money',2,'Lease').
resource_hierarchy(2,'Tangible Resource',1,'Unidentified Tangible Resource').
resource_hierarchy(2,'Tangible Resource',2,'Identified Tangible Resource').

resource_hierarchy(3,'Identified Tangible Resource',1,'Raw Materials Inventory (identified)').
resource_hierarchy(3,'Identified Tangible Resource',2,'WIP Inventory (identified)').
resource_hierarchy(3,'Identified Tangible Resource',3,'Finished Goods Inventory (identified)').
resource_hierarchy(3,'Identified Tangible Resource',4,'Land').
resource_hierarchy(3,'Unidentified Tangible Resource',1,'Raw Materials Inventory').
resource_hierarchy(3,'Unidentified Tangible Resource',2,'WIP Inventory').
resource_hierarchy(3,'Unidentified Tangible Resource',3,'Finished Goods Inventory').
resource_hierarchy(3,'Unidentified Tangible Resource',4,'Shipping Equipment Inventory').
resource_hierarchy(3,'Other Intangible Resource',1,'Share (stock)').
resource_hierarchy(3,'Other Intangible Resource',2,'Goodwill').
resource_hierarchy(3,'Other Intangible Resource',3,'Informed Customer').
resource_hierarchy(3,'Other Intangible Resource',4,'Machine Services').
resource_hierarchy(3,'Other Intangible Resource',5,'Occupancy Rights').

resource_hierarchy(4,'Amortizable Resource',1,'Building').
resource_hierarchy(4,'Amortizable Resource',2,'Plant').
resource_hierarchy(4,'Amortizable Resource',3,'Vehicle').

/*-----------------------------*/
/* Table 2.4B resource_ev_hierarchy(Pl,Pname,Cseq,Cname).*/

resource_ev_hierarchy(0,'Resource',1,'Shrinking Value Resource').
resource_ev_hierarchy(0,'Resource',2,'Non-amortizable Resource').
resource_ev_hierarchy(0,'Resource',3,'Growing Value Resource').

resource_ev_hierarchy(1,'Shrinking Value Resource',1,'Tangible Resource').
resource_ev_hierarchy(1,'Shrinking Value Resource',2,'Intangible Resource').

Manuf_database - 11
resource_ev_hierarchy(1,'Non-amortizable Resource',1,'Cash').
resource_ev_hierarchy(1,'Non-amortizable Resource',2,'Cash at Bank, Current A/C').
resource_ev_hierarchy(1,'Non-amortizable Resource',3,'Trade Debt').
resource_ev_hierarchy(1,'Non-amortizable Resource',4,'Other Debt').
resource_ev_hierarchy(1,'Non-amortizable Resource',5,'Inventory').
resource_ev_hierarchy(1,'Non-amortizable Resource',6,'Land').
resource_ev_hierarchy(1,'Non-amortizable Resource',7,'Share (stock)').

resource_ev_hierarchy(1,'Growing Value Resource',1,'Cash at Bank, Interest Earning').
resource_ev_hierarchy(1,'Growing Value Resource',2,'Debenture').
resource_ev_hierarchy(1,'Growing Value Resource',3,'Lease').
resource_ev_hierarchy(1,'Growing Value Resource',4,'Farm Animals').

resource_ev_hierarchy(2,'Tangible Resource',1,'Building').
resource_ev_hierarchy(2,'Tangible Resource',2,'Plant').
resource_ev_hierarchy(2,'Tangible Resource',3,'Vehicle').
resource_ev_hierarchy(2,'Intangible Resource',1,'Labour Services').
resource_ev_hierarchy(2,'Intangible Resource',2,'Occupancy Rights').
resource_ev_hierarchy(2,'Intangible Resource',3,'Machine Services').
resource_ev_hierarchy(2,'Intangible Resource',4,'Warranty').
resource_ev_hierarchy(2,'Intangible Resource',5,'Goodwill').
resource_ev_hierarchy(2,'Intangible Resource',6,'Informed Customer').

resource_ev_hierarchy(2,'Inventory',1,'Raw Materials Inventory').
resource_ev_hierarchy(2,'Inventory',2,'WIP Inventory').
resource_ev_hierarchy(2,'Inventory',3,'Finished Goods Inventory').
resource_ev_hierarchy(2,'Inventory',4,'Shipping Equipment Inventory').

/*-----------------------------------------------*/
/* Table 2.5 */
composite_resource(Composite_Rno,Date_defined,Component_Rno,Qty_of_component).*
/* Note: Component number unnecessary if not using a relational database */

composite_resource(1030,900101,1020,0.004)/* 1L bottle */
composite_resource(1030,900101,1502,0.01).
composite_resource(1031,900101,1021,1)/* 1L bottles in carton */
composite_resource(1050,900101,1030,12).
composite_resource(1050,900101,1505,0.02).

/*-----------------------------------------------*/
Table 2.6 agent(Agent_id,Agent_name,Address).
agent('A','Adam','Wherever').

Tables 2.7 and 2.8 see end of REE tables.

Table 2.9 tied_asset_obligation(ACno1,ACno2). e.g. leased asset.
tied_asset_obligation(104,290).
tied_asset_obligation(106,291).

Table 2.14 share_stock(Rno,Nominal_value,Date_incorporated).
share_stock(1400,2,860101).

Table 3.15 expected_value(Rno,Expected_life_in_months,Expected_pattern_of_benefits,Expected_residual_value).
expected_value(1100,96,uniform,8000).
expected_value(1101,108,'Sum of years digits',10000).
expected_value(1102,60,uniform,2000).
expected_value(1103,48,uniform,4000).
expected_value(1200,120,uniform,0).
expected_value(1500,0,uniform,0).
expected_value(1501,0,uniform,0).
expected_value(1502,0,uniform,0).
expected_value(1503,0,uniform,0).
expected_value(1701,24,nonuniform,0).

Table 3.16 growing_value_resource(Rno,Expected_life_in_months,Annual_growth_rate,Growth_type).
growing_value_resource(1003,'No Limit',0.045,'Compound, monthly').
growing_value_resource(1300,60,0.07,'Simple').
growing_value_resource(1301,48,0.08,'Simple').
growing_value_resource(1601,48,0.12,'Compound, monthly').
growing_value_resource(1602,48,0.12,'Compound, monthly').

Table 3.17 department_responsible_for_manufacture(Rno,Department).
department_responsible_for_manufacture(1030,'Moulding').
department_responsible_for_manufacture(1031,'Moulding').
department_responsible_for_manufacture(1050,'Packing').
department_responsible_for_manufacture(1051,'Packing').
department_responsible_for_manufacture(1053,'Packing').
/* Table 3.18 production_step_input_AOac(AOac_no,Job_no,Step_no).*/
production_step_input_AOac(410,122,2).
production_step_input_AOac(411,122,2).
production_step_input_AOac(412,122,2).
production_step_input_AOac(420,123,1).
production_step_input_AOac(421,123,1).
production_step_input_AOac(422,123,1).
production_step_input_AOac(430,123,2).
production_step_input_AOac(431,123,2).
production_step_input_AOac(432,123,2).

/* Table 2.7 eh(Eno,Date,Etype,Doc_type,Doc_no,Agent_id,Narrative).*/
eh(0,900601,'Pay Factory Rent',x,x,x,'Pay Factory Rent').
rc(0,1,70,1,1000).
rc(0,2,1,-1000,_,_).

/* Table 2.8 rc(Eno,Line_no,AOacno,Qty,Value).*/
rc(1,1,31,500,20100).
rc(1,2,32,2000,440).
rc(1,3,1,-20540,_,_).

/* Table 3.19 production_step_output_AOac(AOac_no,Job_no,Step_no).*/
production_step_output_AOac(419,122,2).
production_step_output_AOac(429,123,1).
production_step_output_AOac(439,123,2).

/* Table 2.7 eh(Eno,Date,Etype,Doc_type,Doc_no,Agent_id,Narrative).*/
eh(1,900601,'Cash Purchase',x,x,x,'Purchase of Raw Materials').
rc(1,1,31,500,20100).
rc(1,2,32,2000,440).
rc(1,3,1,-20540,_,_).

/* Table 3.19 production_step_output_AOac(AOac_no,Job_no,Step_no).*/
production_step_output_AOac(419,122,2).
production_step_output_AOac(429,123,1).
production_step_output_AOac(439,123,2).
eh(6,900607,'Record Direct Labour','Work Ticket',11,self,'24 hrs factory packing labour on Job 123').
rc(6,1,432,24,__).
rc(6,2,511,-24,__).

eh(7,900607,'Transfer to Finished Goods','Move Ticket',11,self,'Transfer 820 cartons of 1L bottles to Finished Goods').
rc(7,1,511,820,__).
rc(7,2,439,-820,__).

eh(8,900607,'Return to RM Store','Return to Store Ticket',1,self,'Return of 1 20kg bag of pellets to store').
rc(8,1,420,-1,__).
rc(8,2,31,1,__).

eh(9,900607,'Production Summary','Foreman's Report',1,self,'Completion of production of Job 123 - Moulding and Packing').
rc(9,1,449,1,10000,__).
rc(9,2,420,-39,__).
rc(9,3,421,79,__).
rc(9,4,439,820,__).
rc(9,5,430,-10003,__).
rc(9,6,431,-825,__).
rc(9,7,432,-24,__).

eh(10,900606,'Recognize Idle Time','Foreman's Report',1,self,'Idle time Job 123').
r(10,1,512,4,__).
r(10,2,510,-4,__).
r(10,3,513,11,__).
r(10,4,511,-11,__).

eh(11,900606,'Record Factory Wages','Time Sheets',all,self,'Factory wages, employees M,N,Q').
r(11,1,510,83,445,). 
r(11,2,511,35,160,). 
r(11,3,250,220,-50, '/' employee M, 41 hours, 1 lost time ')
r(11,4,251,225,-50, '/' employee N, 42 hours, 3 lost time ')
r(11,5,252,-160,-50, '/' employee Q, 35 hours, 11 lost time'/

/*-----------------------------------------------*/

/* Table 4.1 map AOac to GLac(AOac,GLac).*/
/*Mapping from Asset/Obligation Account to General Ledger A/C */

map_AOac_to_GLac(1,2000). 
map_AOac_to_GLac(2,2001). 
map_AOac_to_GLac(11,2010). 
map_AOac_to_GLac(12,2011). 
map_AOac_to_GLac(31,2021). 
map_AOac_to_GLac(32,2022). 
map_AOac_to_GLac(51,2071). 
map_AOac_to_GLac(52,2072). 
map_AOac_to_GLac(53,2073). 
map_AOac_to_GLac(70,2090). 
map_AOac_to_GLac(72,2090). 
map_AOac_to_GLac(73,2050). */
map_AOac_to_GLac(101,2100).
map_AOac_to_GLac(102,2101).
map_AOac_to_GLac(103,2102).
map_AOac_to_GLac(104,2103).
map_AOac_to_GLac(105,2120).
map_AOac_to_GLac(106,2104).
map_AOac_to_GLac(120,2200).

map_AOac_to_GLac(201,2300).
map_AOac_to_GLac(202,2301).
map_AOac_to_GLac(203,2302).
map_AOac_to_GLac(204,2303).
map_AOac_to_GLac(220,2350).
map_AOac_to_GLac(221,2351).
map_AOac_to_GLac(250,2310).  
map_AOac_to_GLac(252,2311).  
map_AOac_to_GLac(253,2311).  
map_AOac_to_GLac(255,2311).
map_AOac_to_GLac(257,2310).
map_AOac_to_GLac(260,2311).
map_AOac_to_GLac(262,2310).
map_AOac_to_GLac(281,2401).  
map_AOac_to_GLac(282,2402).
map_AOac_to_GLac(290,2410).  
map_AOac_to_GLac(301,2500).  
map_AOac_to_GLac(302,2501).

map_AOac_to_GLac(80,2704). /*Advertising Expense*/
map_AOac_to_GLac(81,2707). /*Office Wages Expense*/
map_AOac_to_GLac(262,2703). /*Interest Expense*/

/* Job 123 Steps 1 and 2 */
map_AOac_to_GLac(420,2040).
map_AOac_to_GLac(421,2041).
map_AOac_to_GLac(429,2040).
map_AOac_to_GLac(429,2041).
map_AOac_to_GLac(429,2811).
map_AOac_to_GLac(430,2045).
map_AOac_to_GLac(431,2045).
map_AOac_to_GLac(432,2046).
map_AOac_to_GLac(439,2045).
map_AOac_to_GLac(439,2046).
map_AOac_to_GLac(439,2821).

map_AOac_to_GLac(500,2046).
map_AOac_to_GLac(510,3000).
map_AOac_to_GLac(511,3000).
map_AOac_to_GLac(512,2810).
map_AOac_to_GLac(513,2820).

/* Table 4.2 map_AC_concept_to_GLac(Accounting_concept,GLac).*/
/* Mapping Accounting concepts to General Ledger A/Cs */

map_AC_concept_to_GLac('Sales Revenue',2700).
map_AC_concept_to_GLac('Profit and Loss',2710).
map_AC_concept_to_GLac('Cost of Sales',2711).
map_AC_concept_to_GLac('Retained Earnings',2600).
map_AC_concept_to_GLac('Gearing Gain',2603).
map_AC_concept_to_GLac('Real Holding Gain',2604).
map_AC_concept_to_GLac('CCA Interindex Gain',2605).
map_AC_concept_to_GLac('Opening Balance Fudge',2606).
map_AC_concept_to_GLac('Provision for Doubtful Debts',2712).
map_AC_concept_to_GLac('Dividend Received',2713).
map_AC_concept_to_GLac('Discount Revenue',2701).
map_AC_concept_to_GLac('Discount Expense',2705).
map_AC_concept_to_GLac('Share Premium Reserve',2602).
map_AC_concept_to_GLac('Share Discount',not defined in GL).
map_AC_concept_to_GLac('Lease Expense',2706).
map_AC_concept_to_GLac('Labour Clearing',3000).
map_AC_concept_to_GLac('Materials Price Variance',3001).
map_AC_concept_to_GLac('Labour Rate Variance',3002).

/* Table 4.3 map_etype_to_eproc(Event_type,event_handler). */

map_etype_to_eproc('Credit Sale',credit_sale).
map_etype_to_eproc('Cash Sale',credit_sale).
map_etype_to_eproc('Sale of Amortizable Asset',sale_amortizable_asset).
map_etype_to_eproc('Open Asset and Obligation Accounts',open_acs).
map_etype_to_eproc('Open Amortizable Asset Account',open_amortizable_asset_acs).
map_etype_to_eproc('Cash Receipt',cash_receipt).
map_etype_to_eproc('Cash Payment',cash_payment).
map_etype_to_eproc('Pay Supplier',cash_payment).
map_etype_to_eproc('Raw Materials Issue',manuf_event).
map_etype_to_eproc('Record Direct Labour',manuf_event).
map_etype_to_eproc('Record Factory Wages',manuf_event).
map_etype_to_eproc('Interdepartmental Goods Transfer',manuf_event).
map_etype_to_eproc('Transfer to Finished Goods',manuf_event).
map_etype_to_eproc('Production Summary',production_summary).
map_etype_to_eproc('Return to RM Store',manuf_event).
map_etype_to_eproc('Recognize Idle Time',manuf_event).
map_etype_to_eproc('Issue Shares in Exchange for Factory',issue_shares).
map_etype_to_eproc('Issue Shares',issue_shares).
map_etype_to_eproc('Pay Interest on Debentures',expense_immediately).
map_etype_to_eproc('Pay Advertising Costs',expense_immediately).
map_etype_to_eproc('Declare Dividend',declare_dividend).
map_etype_to_eproc('Receive Dividend',receive_dividend).
map_etype_to_eproc('Sell Goods Under Warranty',no_journal_entry_required).

/* yet to be coded: */

Manuf_database - 17
map_etype_to_eproc('Lease Tangible Asset', lease_tangible_asset).
map_etype_to_eproc('Lease Payment', lease_payment).
map_etype_to_eproc('Interest Payment', interest_payment).

/* simple */
map_etype_to_eproc('Credit Purchase', simple).
map_etype_to_eproc('Cash Purchase', simple).
map_etype_to_eproc('Buy Tangible Asset', simple).
map_etype_to_eproc('Receive Goods', simple).
map_etype_to_eproc('Receive Supplier Invoice', simple).
map_etype_to_eproc('Pay Employee', simple).
map_etype_to_eproc('Issue Debentures', simple).
map_etype_to_eproc('Pay Factory Rent', simple).
map_etype_to_eproc('Opening Inventory', simple).

/* no journal entry required */
map_etype_to_eproc('Sales Order', no_journal_entry_required).
map_etype_to_eproc('Purchase Order', no_journal_entry_required).
map_etype_to_eproc('Production Order', no_journal_entry_required).
map_etype_to_eproc('Record Machine Hours', no_journal_entry_required).

/* Table 4.4 unit_standard_cost(Rno, Date_set, Standard_cost). */
unit_standard_cost(1020, 900101, 40.0). /* poly pellets */
unit_standard_cost(1021, 900101, 0.22). /* cartons */
unit_standard_cost(1052, 900101, 30.00). /* carton metal caps */
unit_standard_cost(1502, 900101, 5.00). /* moulding labour */
unit_standard_cost(1503, 900101, 8.00). /* m/c #1; str line depr = $833 p.m. => $5.2/hr */
unit_standard_cost(1505, 900101, 4.00). /* packing labour */
unit_standard_cost(1060, 900101, 10.00). /* pallet, charged out at $10 */

/* Table 4.5 departmental_OH_rate(Department, Date_set, Rate, Basis_of_variation). */
departmental_OH_rate('Moulding', 900101, 8.0, 'dlh').
departmental_OH_rate('Packing', 900101, 2.0, 'dlh').

/* Table 4.6 department_OH_flexible_budget(Dept, Fixed_OH, Var_OH_coefficient, Basis_of_variation). */
department_OH_flexible_budget('Moulding', 1000, 4, 'standard dlh').
department_OH_flexible_budget('Packing', 200, 1, 'standard dlh').

/* Table 4.7 price_index(Series, Date_set, Index_value). */
price_index(g, 900101, 100).
price_index(g, 940101, 120).
price_index(g, 760101, 40).
price_index(g, 900101, 100).
price_index(g, 900201, 102).

Manuf_database - 18
price_index(g,900301,105).
price_index(g,900401,103).
price_index(g,900501,104).
price_index(g,901001,109).
price_index(g,910101,112).

price_index(s,90101,100).
price_index(s,900201,104).
price_index(s,900301,106).
price_index(s,901001,118).
price_index(s,910101,120).

price_index(sv,900101,100).
price_index(sv,901001,118).
price_index(sw,900101,100).
price_index(sw,901001,118).

price_index(sp,760701,100).
price_index(sp,900701,200).

/*
  Table 4.8 resource_price_level_series_mapping(Rtype,Series).
*/

resource_price_level_series_mapping('Cash',g).
resource_price_level_series_mapping('Cash at Bank, Current A/C',g).
resource_price_level_series_mapping('Cash at Bank, Interest Earning',g).
resource_price_level_series_mapping('Trade Debt',g).
resource_price_level_series_mapping('Other Debt',g).
resource_price_level_series_mapping('Debenture',g).
resource_price_level_series_mapping('Lease',g).
resource_price_level_series_mapping('Raw Materials Inventory (identified)',si).
resource_price_level_series_mapping('WIP Inventory (identified)',si).
resource_price_level_series_mapping('Finished Goods Inventory (identified)',si).
resource_price_level_series_mapping('Raw Materials Inventory',si).
resource_price_level_series_mapping('WIP Inventory',si).
resource_price_level_series_mapping('Finished Goods Inventory',si).
resource_price_level_series_mapping('Shipping Equipment Inventory',g).
resource_price_level_series_mapping('Land',g).
resource_price_level_series_mapping('Share (stock)',g).
resource_price_level_series_mapping('Farm Animals',g).
resource_price_level_series_mapping('Building',sp).
resource_price_level_series_mapping('Plant',sp).
resource_price_level_series_mapping('Vehicle',sv).
resource_price_level_series_mapping('Labour Services',sw).
resource_price_level_series_mapping('Occupancy Rights',g).
resource_price_level_series_mapping('Machine Services',g).
resource_price_level_series_mapping('Warranty',g).
resource_price_level_series_mapping('Goodwill',g).
resource_price_level_series_mapping('Informed Customer',g).

/* ACCOUNTING "GIVEN" INFORMATION AT SYSTEM STARTUP */

/* Table 5.1 gl_opening_balance(PLA_type,LIFO_FIFO,GL_ACno,Balance). */

gl_opening_balance('Historical Cost',fifo,2602,-2000).
gl_opening_balance('Historical Cost',fifo,2700,-5000).
gl_opening_balance('Historical Cost',fifo,2711,3000).
gl_opening_balance('Historical Cost',fifo,2800,200).

gl_opening_balance('Current Purchasing Power',fifo,2711,3118.81).
gl_opening_balance('Current Purchasing Power',fifo,2800,205.88).

gl_opening_balance('Current Cost Accounting',fifo,2602,-2500).
gl_opening_balance('Current Cost Accounting',fifo,2603,212.41).
gl_opening_balance('Current Cost Accounting',fifo,2604,-65.03).

/* above allows for net -252.23 from adjusts to ed() entries ==> -317.26 */
gl_opening_balance('Current Cost Accounting',fifo,2700,-5097.09).
gl_opening_balance('Current Cost Accounting',fifo,2711,3178.18).
gl_opening_balance('Current Cost Accounting',fifo,2800,211.54).

gl_opening_balance('Current Cost Accounting',fifo,2602,-2000).
gl_opening_balance('Current Cost Accounting',fifo,2700,-5000).
gl_opening_balance('Current Cost Accounting',fifo,2711,3000).
gl_opening_balance('Current Cost Accounting',fifo,2800,200).

/* Table 5.2 cca_account_series(PLA_type,GL_ACno,Series). */

cca_account_series('Current Cost Accounting',2602,g).
cca_account_series('Current Cost Accounting',2603,g).
cca_account_series('Current Cost Accounting',2604,sv).
cca_account_series('Current Cost Accounting',2700,g).
cca_account_series('Current Cost Accounting',2711,g).
cca_account_series('Current Cost Accounting',2800,sv).
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent(Agent_id,Agent_name,Address)</td>
<td>13</td>
</tr>
<tr>
<td>allocation_of_OH(From_GLac,To_GLac,Proportion)</td>
<td>6</td>
</tr>
<tr>
<td>ao_account(ACno,Description,Class,Rno,Agent_id,Controller)</td>
<td>7</td>
</tr>
<tr>
<td>ao_hierarchy(PI,Phame,Cseq,Cname)</td>
<td>8</td>
</tr>
<tr>
<td>cca_account_series(PLA_type,GL_ACno,Series)</td>
<td>20</td>
</tr>
<tr>
<td>chart_of_accounts(GLacno,description,class)</td>
<td>2</td>
</tr>
<tr>
<td>composite_resource(Composite_Rno,Date_defined,Component_Rno,Qt)</td>
<td>12</td>
</tr>
<tr>
<td>department_GL_variance_account(Department,Variance_ac_descrit)</td>
<td>6</td>
</tr>
<tr>
<td>department_OH_flexible_budget(Dept,Fixed_OH,Var_OH_coefficient)</td>
<td>18</td>
</tr>
<tr>
<td>department_responsible_for_manufacture(Rno,Department)</td>
<td>13</td>
</tr>
<tr>
<td>departmental_OH_rate(Department,Date_set,Rate,Basis_of_variability)</td>
<td>18</td>
</tr>
<tr>
<td>eh(Eno,Date,Etype,Doc_type,Doc_no,Agent_id,Narrative)</td>
<td>14</td>
</tr>
<tr>
<td>expected_value(Rno,Expected_life_in_months,Expected_pattern_of)</td>
<td>13</td>
</tr>
<tr>
<td>expense_ac_for_growing_liability(GL_ACno,Expense_ACno)</td>
<td>5</td>
</tr>
<tr>
<td>gi_amortization(GL_ACno,Expense_ACno)</td>
<td>5</td>
</tr>
<tr>
<td>gi_depreciation(Asset_ACno,Accum_Depr_ACno,Depr_Exp_ACno)</td>
<td>5</td>
</tr>
<tr>
<td>gi_opening_balance(PLA_type,LIFO_FIFO,GL_ACno,Balance)</td>
<td>20</td>
</tr>
<tr>
<td>giAC_hierarchy(P_seq,P_name,C_seq,C_name)</td>
<td>4</td>
</tr>
<tr>
<td>growing_value_resource(Rno,Expected_life_in_months,Annual_growth)</td>
<td>13</td>
</tr>
<tr>
<td>jed(Eno,Line_no,JE_ACno,Amount)</td>
<td>5</td>
</tr>
<tr>
<td>jeh(Eno,Date,Eno,Narration)</td>
<td>5</td>
</tr>
<tr>
<td>map_AC_concept_to_GLac(Accounting_concept,GLac)</td>
<td>16</td>
</tr>
<tr>
<td>map_AOac_to_GLac(AOac,GLac)</td>
<td>15</td>
</tr>
<tr>
<td>map_etype_to_eproc(Event_type,event_handler)</td>
<td>17</td>
</tr>
<tr>
<td>price_index(Series,Date_set,Index_value)</td>
<td>18</td>
</tr>
<tr>
<td>production_step_GLacs(Job_no,Step_no,M_GLac,L_GLac,OH_Incurred)</td>
<td>7</td>
</tr>
<tr>
<td>production_step_input_AOac(AOac_no,Job_no,Step_no)</td>
<td>14</td>
</tr>
<tr>
<td>production_step_output_AOac(AOac_no,Job_no,Step_no)</td>
<td>14</td>
</tr>
<tr>
<td>rc(Eno,Line_no,AOacno,Qty,Value)</td>
<td>14</td>
</tr>
<tr>
<td>resource(Rno,Description,Unit_of_measure,Rtype)</td>
<td>10</td>
</tr>
<tr>
<td>resource_ev_hierarchy(Pl,Phame,Cseq,Cname)</td>
<td>11</td>
</tr>
<tr>
<td>resource_hierarchy(Pl,Phame,Cseq,Cname)</td>
<td>11</td>
</tr>
<tr>
<td>resource_price_level_series_mapping(Rtype,Series).*/</td>
<td>19</td>
</tr>
<tr>
<td>share_stock(Rno,Nominal_value,Date_incorporated)</td>
<td>13</td>
</tr>
<tr>
<td>tied_asset_obligation(ACno1,ACno2)</td>
<td>13</td>
</tr>
<tr>
<td>unit_standard_cost(Rno,Date_set,Standard_cost)</td>
<td>18</td>
</tr>
<tr>
<td>wip_GLac_class(GL_ACno,WIP_class)</td>
<td>6</td>
</tr>
</tbody>
</table>

Manuf_database - 21
Bibliography


bib - 3


(1907). The Rate of Interest, New York: Macmillan.


bib - 10


bib - 11


____ (1952b). *Structure of Accounting Theory*, AAA Monograph No.5, AAA.


Pacioli, Luca (1496). Summa de Arithmetica, Geometria, Proportioni et Proportionalita, Venice, 1496.


bib - 18


An architecture for computer-based accounting information systems


Publication Status:
Unpublished

Persistent Link:
http://hdl.handle.net/11343/37822

File Description:
An architecture for computer-based accounting information systems

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.