9. AN EVENTS ACCOUNTING FOUNDATION FOR DSS IMPLEMENTATION*

Eric L. Denna and William E. McCarthy
Department of Accounting
Michigan State University
East Lansing, Mich. 48824, U.S.A.

ABSTRACT

This paper describes an implementation of an events accounting system which was specifically designed to facilitate the use of its disaggregated data in support of various intelligence, design, and choice decisions in a manufacturing environment. The system also models the processing of normal accounting data and reports, but it does so by materializing chart of account classifications exclusively with procedures (as opposed to conventional general ledger systems which rely primarily on declarative features). The particular implementation described herein uses a relational data base foundation, and its decision support is effected with a variety of microcomputer facilities, such as spreadsheets and graphics. The paper concentrates first on describing the derivation of data-modelled relations in accordance with an REA accounting framework. It then proceeds with a description of the DSS framework used and the correspondence of various components of this implementation with elements of that framework. The paper concludes with an assessment of problems encountered and an enumeration of future plans for more complete integration of accounting systems with marketing, materials, and logistics planning systems.

1. INTRODUCTION

According to Keen [1976], the concept of decision support systems (DSS) is based on several assumptions about the role of the computer in effective decision making [Davis and Olson, 1985, pp. 368-369]:

*Support for the computer implementation described in this paper was provided by The Touche Ross Foundation through its Aid to Accounting Education Program.

NATO ASI Series, Vol. F31
Decision Support Systems: Theory and Application
Edited by C.W. Holsapple and A.B. Whinston

1. The computer must support the manager but not replace his or her judgement. It should therefore neither try to provide the "answers" nor impose a predefined sequence of analysis.

2. The main payoff of computer support is for semi-structured problems, where parts of the analysis can be systematized for the computer, but where the decision maker's insight and judgement are needed to control the process.

3. Effective problem solving is interactive and is enhanced by a dialog between the user and the system. The user explores the problem situation using the analytic and information-providing capabilities of the system as well as human experience and insights.

The DSS idea of interactive support for semistructured decision makers is in sharp contrast to what most managers would consider the orientation of computerized accounting systems where the emphasis traditionally has been on the provision of narrow, historical data for precisely defined needs. However, much of the information accountants gather does provide at least partial support for unstructured decisions, and we intend to propose in this paper that the extent of that support could be broadened if the accounting systems in question could be designed differently. More specifically, we will present a small prototype of a semantically modelled accounting information system which presently is being implemented with the relational data base system Knowledge Manager [Holsapple and Whinston, 1984]. Knowledge Manager (or KMAN as we will refer to it from this point on) is actually much more comprehensive in scope than most data base systems, because it contains integrated capabilities for spreadsheet processing, text editing, graphics, and forms management as well as capabilities for a complete range of both navigational and specificational [Tsichritzis and Lochovsky, 1982] data base language operations. Holsapple and Whinston [1984] have proposed KMAN as an ideal candidate for the problem processing system (PPS) component of a generalized DSS, and we will be looking at those capabilities in this paper in the context of a wide-ranging accounting example.

The remainder of this paper is organized as follows. In the next section, we will be looking at the conceptual structure of an accounting information system implemented on KMAN. Following that, we will explain how the set of base relations used in this prototype was derived with the REA model [McCarthy, 1982] as a guide and how those relations were used to materialize a normal set of financial statement accounts. At that point, we will have explained a working accounting system, and we will proceed by exploring the use of this small prototype in the development of some sample DSS. In our conclusion, we will comment on the future directions of this implementation.

2. CONCEPTUAL STRUCTURE OF AN ACCOUNTING INFORMATION SYSTEM

In their book on management information systems, Davis and Olson define the conceptual structure of an information system (see, for example, Figure 1) as follows:

> ... a federation of functional subsystems, each of which is divided into four major information processing components: transaction processing, operational control information system support, managerial control information system support, and strategic planning information system support. Each of the functional subsystems of the information system has some unique data files which are used only by that subsystem. There are also files which need to be accessed by more than one application and need to be available for general retrieval. These files are organized into a general database managed by a database management system.
>
> A further amplification of the structure is the introduction of common software. In addition to application programs written especially for each subsystem, there are common applications which serve multiple functions. Each subsystem has linkages to those common applications. There are also many analytical and decision models that can be used by many applications. These form the model base for the information system. [Davis and Olson, 1985, pp. 45-47]

The prototype accounting information system and decision support facility which we are building with KMAN can be explained well with the terms of the Figure 1 structure. Applicable components are outlined below.

<u>The application software</u> for each subsystem consisted of programs written in both procedural and relational languages. In contrast to most batch oriented accounting systems, our transaction processing programs were written for on-line entry using the forms definition facility of KMAN. The

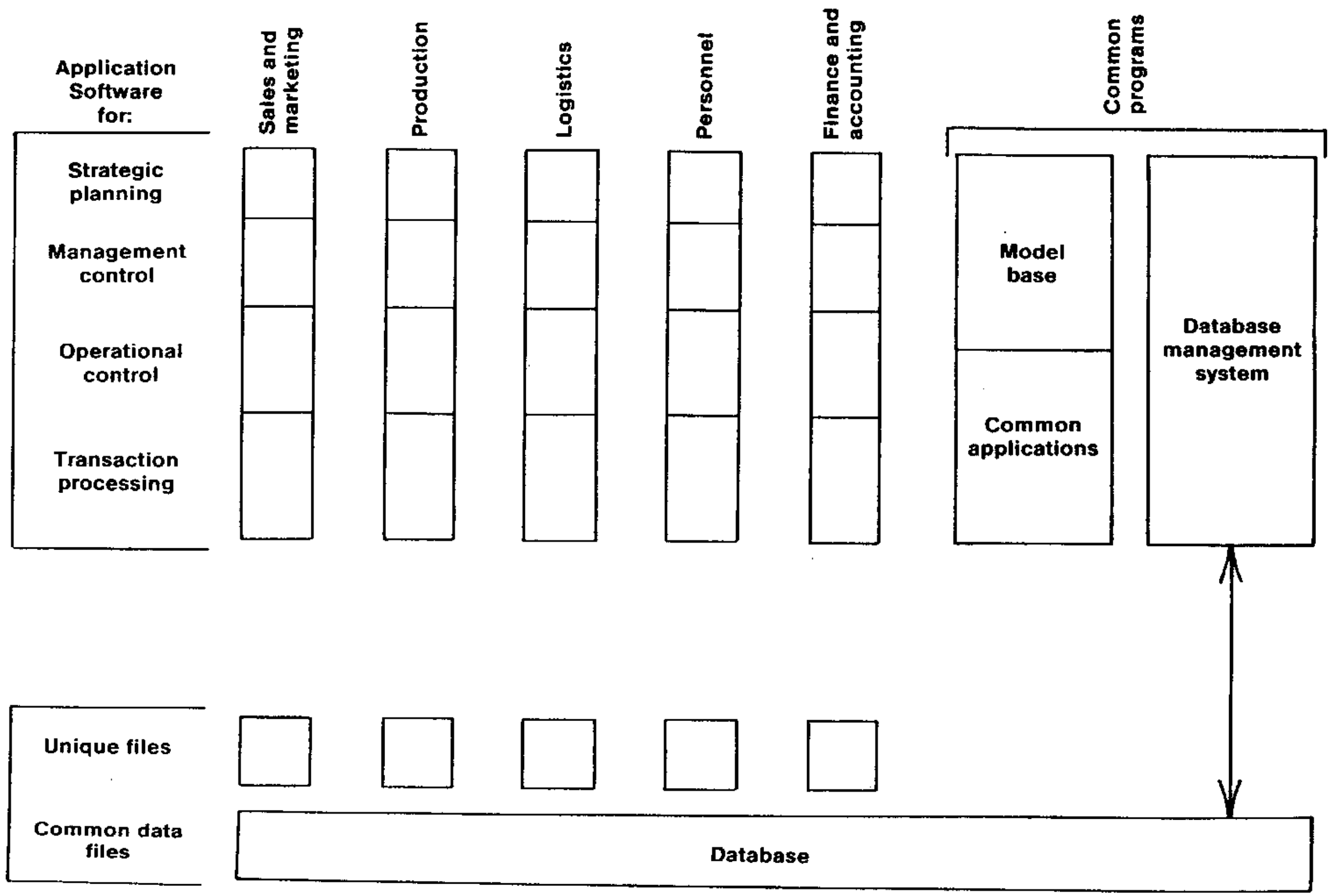

Figure 1. Conceptual Structure of an Information System
(Source: Davis and Olson [1985, p. 46])

operational control, management control, and strategic planning software consisted of spreadsheets, report generators, graphics generators, and ad hoc retrievals written for each area.

The model base consisted of spreadsheet programs or data generation programs (such as time series analysis and correlation or extrapolation techniques) which are used by more than one subsystem.

The common applications were the programs which ranged across the separate subsystems to gather the general ledger data and produce the financial statements.

The data base management system was Knowledge Manager.

The common data files were the relations generated using conceptual data modelling and the REA [McCarthy, 1982, 1984] framework. These are referred to as base tables.

The unique files were the views or "virtual relations" which we derived from the base relations procedurally whenever required.

With the major exception of the base tables, views, and application software needed to materialize the general ledger, we have implemented only part of some of the components mentioned above. We plan to implement everything, but time constraints dictate that the more novel features be addressed first. For example, having once demonstrated KMAN's facility with form-driven input, we simply filled in the rest of the accounting events.

The feature that differentiates the information system prototype illustrated in Figure 1 from a conventional accounting system is the integrated nature of its transaction processing capabilities and its decision support capabilities -- a feature due both to the integrated software used and to the method of data base design employed. The nature of that design process and the closely allied concept of events accounting are explained next.

3. EVENTS ACCOUNTING AND DATA BASE DESIGN WITH THE REA ACCOUNTING MODEL

3.1. Events Accounting

In very general terms, the concept of events accounting [Sorter, 1969; McCarthy, 1981] indicates a movement:

- away from the idea of an accountant as a person who observes the economic activities of an enterprise, records those observations as data, and then summarizes and interprets that data for interested parties, and
- toward the idea of an accountant as a person who observes and records economic activity and then simply leaves the resulting data to be interpreted by interested users.

As many computer scientists will realize, this events approach corresponds very closely to a movement in fields such as artificial intelligence and data base management called conceptual modelling [Sowa, 1984; Brodie et al., 1984; Kent, 1978]. Conceptual modelling has as its fundamental tenet the realization that the closer one can make the machine conceptualization of reality to the human conceptualization, the more useful and stable will be any information system built on that machine conceptualization. In data base management, the growing importance of conceptual modelling is reflected in the increasing attention given to semantic and infological data models [Tsichritzis and Lochovsky, 1982].

Semantic data models and events accounting systems have the same overall objective in that they both endeavor to construct man-made representations of real world phenomena. That portion of the real world that a particular designer is trying to capture is normally called the object system, and in accounting, this object system is usually a business enterprise. It is very natural then to use data-modelled concepts as the basis for user-interpreted accounting information systems. The most recent and comprehensive of these conceptual accounting frameworks is McCarthy's REA accounting model, and it was this particular framework which we used in our KMAN implementation for derivation of the common data file component of the information system prototype.

3.2. The REA Accounting Model

The declarative structure of the REA accounting model is shown in Figure 2. The individual elements of this structure were derived by McCarthy [1982] via a semantic analysis of chart of account structures in a typical general ledger and via an overall analysis of the writings of accounting theorists such as Ijiri [1975] and Mattessich [1964]. REA components are characterized with Chen's [1976], McCarthy's [1979] Entity-Relationship (E-R) semantic data model. Roles of each entity within a relationship are illustrated also.

Figure 2 represents a prototypical instance [Borgida et al., 1984] of the concept of an accounting transaction or event. As the reader can see, each of these events normally requires both an inside and outside agent to act as parties to the transaction, and the event itself normally results in either an inflow or an outflow of resources into an enterprise. If the concept of economic event was instantiated in a retail company with sale, the outside and inside parties could be customer and salesperson respectively. The economic resource involved (in this case an outflow) would be some kind of inventory. Each event which results in an outflow of resources to the enterprise is required to be coupled to an event which results in an inflow and vice-versa. The matching event in the case of the sale transaction would be a cash receipt which the customer would normally remit at a later date. The constraint and procedural aspects of the REA framework are explored in other papers [Gal and McCarthy, 1983, 1985b, 1985c; McCarthy, 1984].

3.3. The Enterprise Modelling Process

For our KMAN implementation, we chose a simple manufacturing enterprise as the object system. This enterprise buys raw materials from various vendors and then converts those raw materials into finished goods in a manufacturing process which uses extensive amounts of employee labor and which needs certain machines and other factory facilities. The finished goods are sold to customers by salespeople. This particular enterprise has a number of other employees, and it incurs a number of other

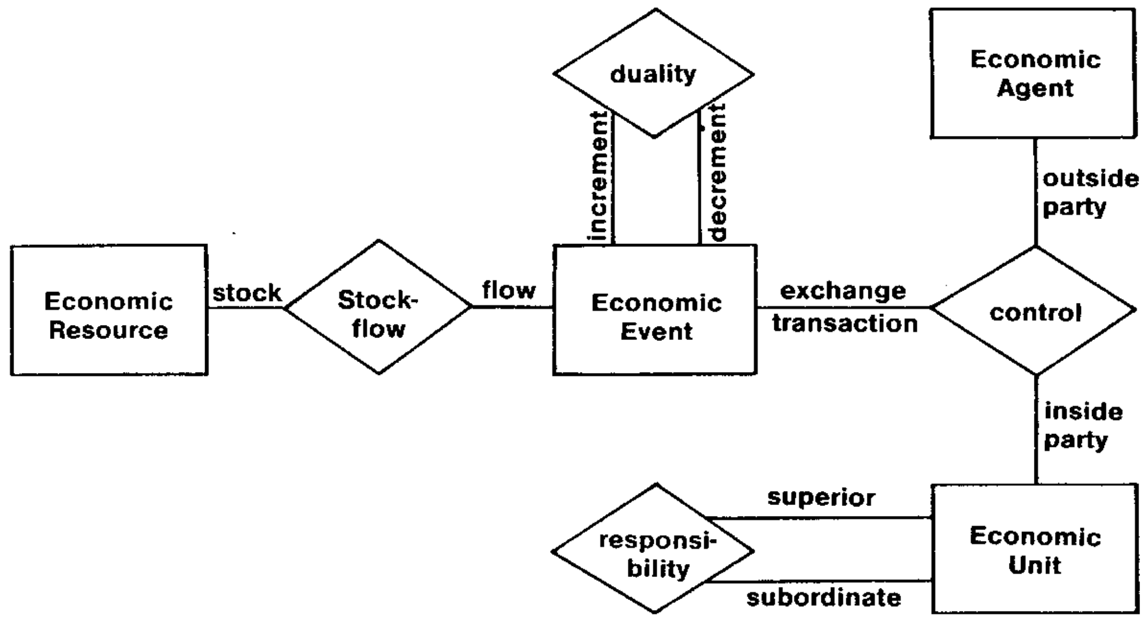

Figure 2. The REA Accounting Model
(Source: Adapted from McCarthy [1982, p. 564])

operating expenses such as rent and advertising.

The process of designing a data base begins with requirements analysis [Lum et al., 1979]. In our events system, this was a relatively uncomplicated step because we deliberately circumscribed the set of economic phenomena to be modelled in an effort to get the prototype implemented. The set of transactions which we used as a test case are given in Appendix 1 of [Denna and McCarthy, 1985].

After requirements analysis, data base design proceeds with the conceptual modelling of the enterprise. This process can be divided into the two steps explained below: view modelling and view integration.

View modelling takes the list of data elements which either an individual program or an individual decision maker needs for a particular task, and it then characterizes those data elements in terms of a particular semantic model. This list of data elements could come from documentation of the existing system (such as a data store on a data flow diagram [Demarco, 1979]), or it could be derived using other requirements elicitation techniques (such as business systems planning or critical success factor identification [Martin, 1982]).

In an REA-modelled environment, the list of data elements in a view is interpreted semantically in terms of the prototypical event instance illustrated in Figure 2. We have included an example of such an interpretation in Figure 3 for a view of labor operation processing. The economic events, agents, and resources along with their accompanying attributes are recast in an entity-relationship diagram. We have augmented in Figure 3 the basic E-R diagramming methods with conventions introduced by Atzeni et al. [1983] and Howe [1983] which illustrate attributes of entities as circles (filled-in circles are keys) and participation by entities in a relationship as either obligatory (dot inside the box) or non-obligatory (dot outside the box).

View integration takes all of the individual E-R diagrams derived above and combines them into one enterprise model. This is a very complicated step which necessitates much detailed work

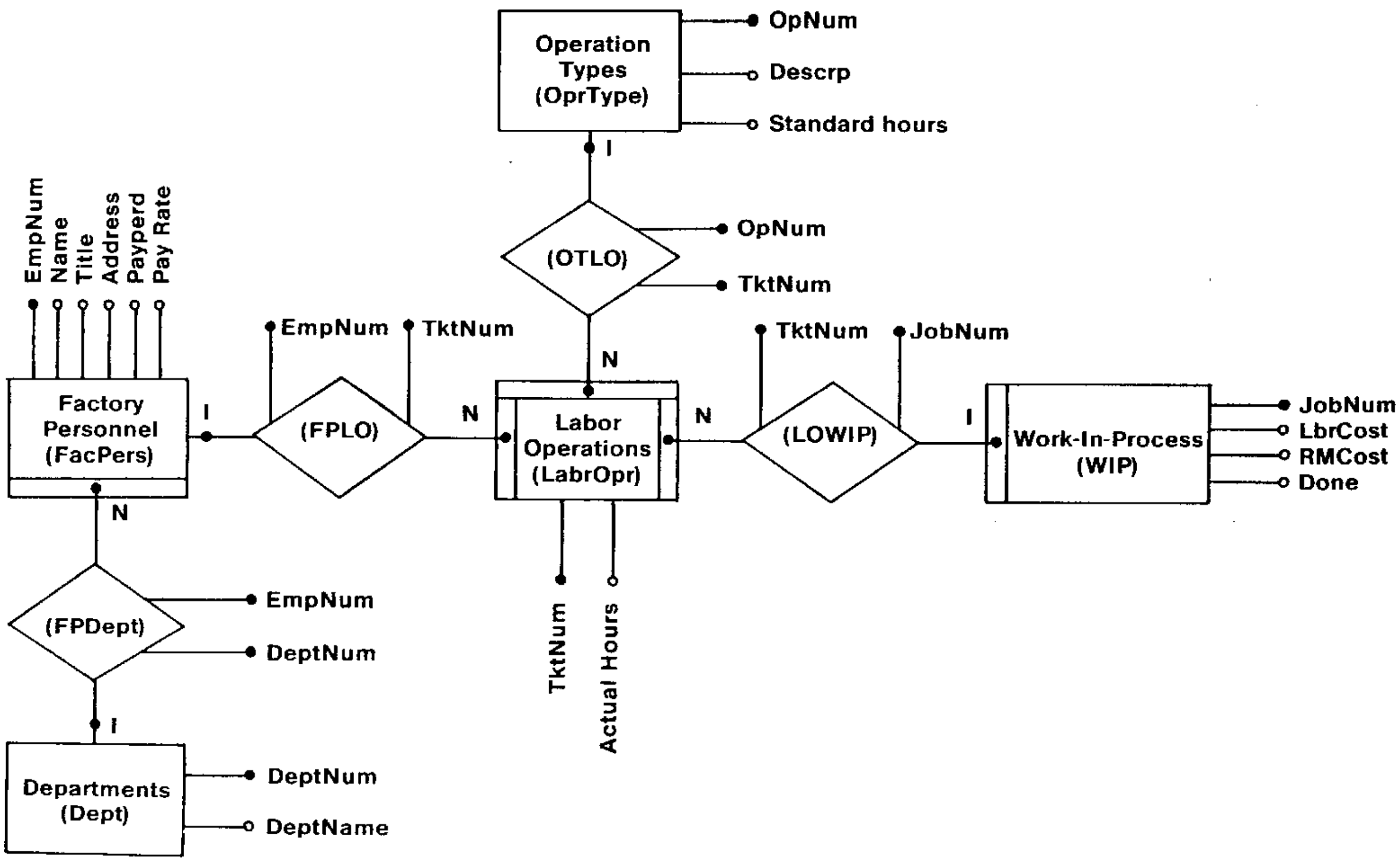

Figure 3. Instantiation of REA Template

such as the resolution of naming and ownership inconsistencies. However, for our purposes in this paper, we can describe it at a more abstract level as a process of linking individual event templates together. For example, the outflow event sale could be linked to the inflow event received as economic compensation for it: cash receipt. Additionally, since sale is an outflow of finished goods inventory, it would become linked to the inflow event for that resource which in this case would be a job operation that transfers work-in-process to finished goods. An example E-R model which illustrates some of this integration is given as Figure 4. After much of this linking is finished, an additional step is performed which constructs generalization hierarchies for those separate entity sets which need to be considered collectively for some type of information processing. For example in our manufacturing object system, we kept separate classes of employees and inventory as different base tables, but we combined them together into one superset for processing such as payroll [Smith and Smith, 1977].

View modelling and integration are much more complex operations than the brief descriptions given above would indicate, but the discussions do give an appreciation for the use of the REA framework in these processes. Formulation of the data base system proceeds from this point with implementation design and physical data base design, two additional steps for which we also will limit our discussions to germane features of our particular KMAN implementation.

Implementation design entails tailoring the semantic model derived in view integration to the particular data base system to be used. In our case, this meant deriving relations from E-R models. This was very straightforward for us, because we were dealing with a limited quantity of data. In a more realistic events setting however, complex issues such as temporal aggregation of data and non-implementation of certain entities and relationships would have to be addressed. Additional organizational implementation issues such as the division of the enterprise model into subject data bases would also have to be considered [Martin, 1982]. A list of the relations used in our

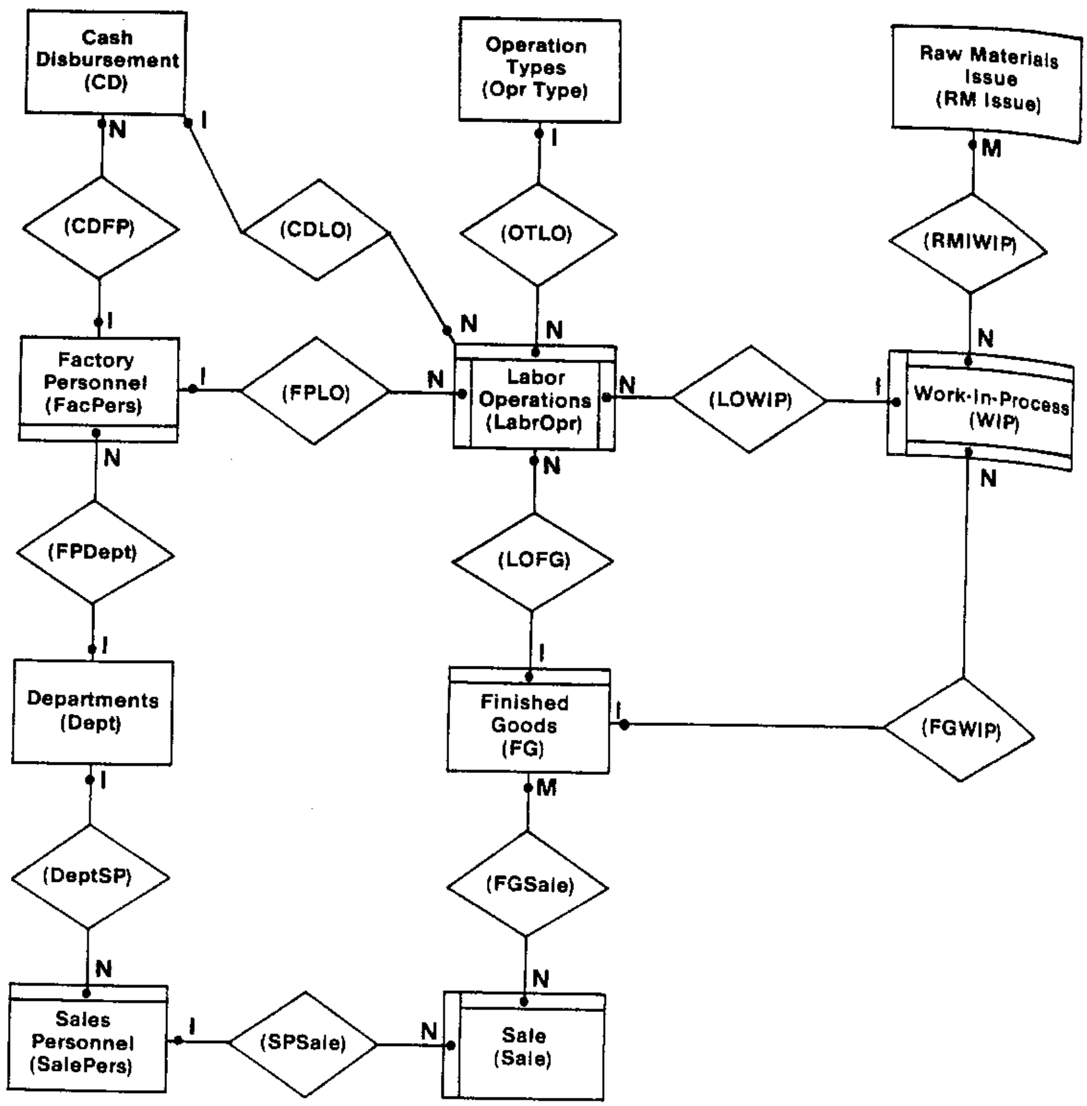

Figure 4. Integration of REA Views in Manufacturing

implementation is included in Appendix 2 of [Denna and McCarthy, 1985].

Physical data base design involves choices such as the selection of access paths and indices. Most of the issues here are not relevant to our discussion.

At this point, we are finished describing our enterprise modelling process. We are reminded again that a much more thorough data modelling effort would be needed to deal with the complexity of an actual enterprise [Martin, 1983, p. 688], but the descriptions here sufficiently characterize our prototype.

3.4. Materialization of Account Balances and Statements

In terms of the conceptual structure of an information system illustrated in Figure 1, the data base design process described above would produce the common data files or the base tables implemented in KMAN. In this section of the paper, we will explain our use of the unique files (which were views or virtual relations) and our use of the common applications (which were the programs to gather the general ledger and produce the financial statements).

The concept of events accounting requires that the corporate store of accounting data be kept in a conceptually modelled data base which captures as naturally as possible the essence of the enterprise being tracked. Therefore predefined aggregations of data are to be avoided as much as possible, because they are biased toward the perspective of the designer or decision maker doing the aggregation. Again, this is a philosophy that events accounting shares with semantic modelling, although it is certainly true that some interpretations and representation trade-off decisions will always have to be made.

In our accounting prototype system, we decided to operationalize this philosophy by relegating (as much as possible) any processed data to views [Date, 1981] or virtual relations. Thus, we tried to define base tables only for those objects which seemed to model directly some aspect of the real world. We realize that this is an impossible objective given the problems of "relativism" [Hammer and McLeod, 1981; Winograd, 1975] (i.e., one person's entity is another's relationship is another's attribute is another's procedure ... etc.). However, as a philosophy of design, we believe that this approach has merit because it always causes a designer to look for the solution which has the closest correspondence to the underlying object system.

KMAN does not provide a view definition facility explicitly (neither do most other data base systems), so we had to implement our views with a technique that invoked a procedure which first deleted the view's extension (that is, the occurrence data or the actual rows in a data base table) and then repopulated that extension with another procedure. This same technique was used by Gal and McCarthy in a QBE data base implementation, and readers interested in more detail can refer to [Gal and McCarthy, 1985a].

The structure chart or procedure hierarchy which we used for the one application which ranged across all of our functional areas is shown in Figure 5. This application materialized the chart of accounts for our sample enterprise, and its final output was a view (or virtual table) which listed all of our account balances. The structure of this procedure hierarchy was based on one developed by McCarthy [1984] from the conceptual accounting framework project of the Financial Accounting Standards Board [1980]. With the events accounting emphasis on disaggregate and uninterpreted data, all of the account balances used for our general ledger were always current as of the last transaction occurrence, because they were materialized only as needed. After materialization, they were either displayed or passed to KMAN's form facility to be printed in the form of financial statements such as an income statement or a balance sheet. Some of the programs used in the general ledger derivation are reproduced in Appendix 3 of [Denna and McCarthy, 1985].

In addition to the general ledger materialization programs, there were other KMAN procedures written to perform accounting functions. There were form-oriented transaction entry procedures, and there were programs written to materialize generalization

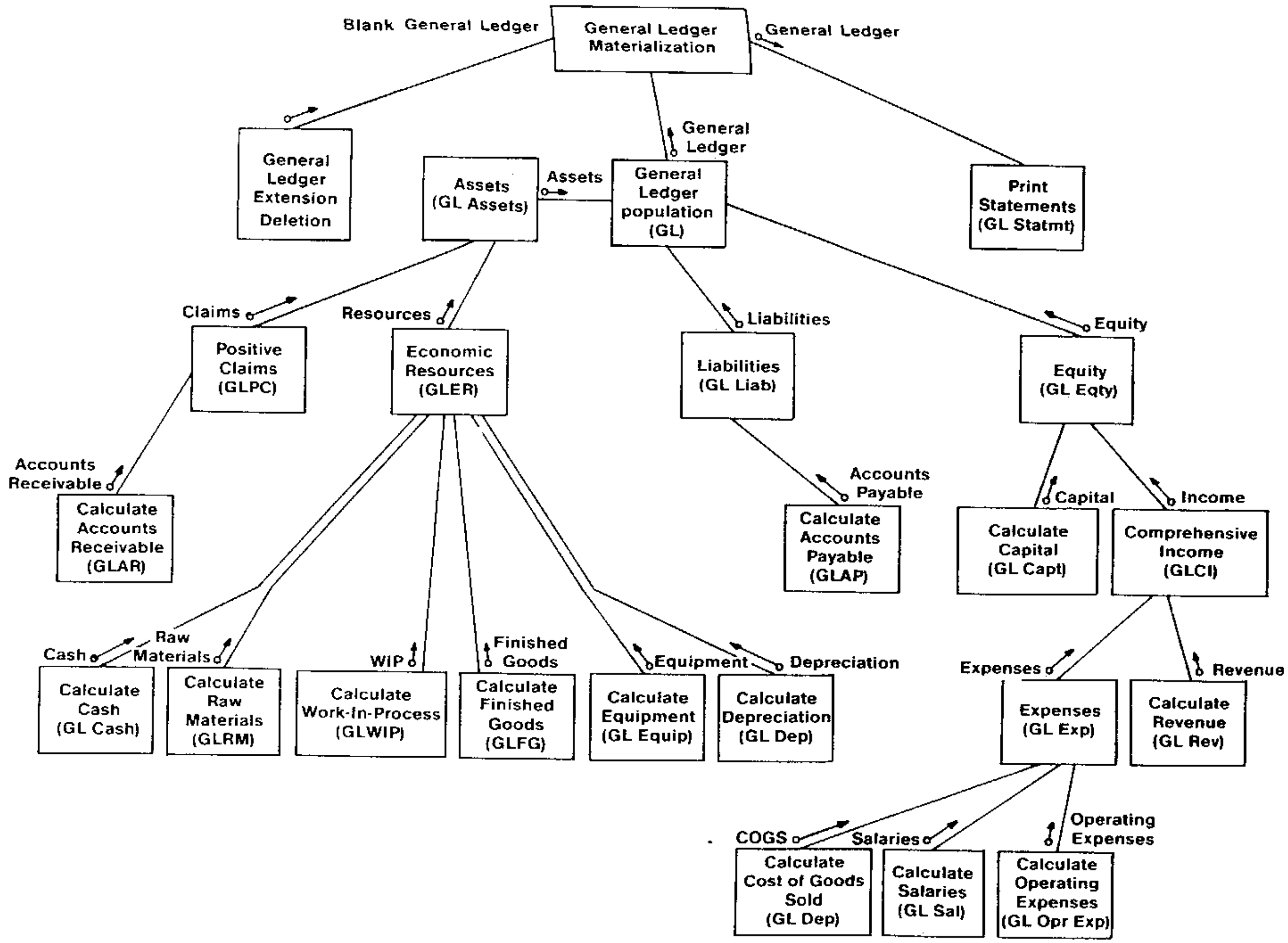

Figure 5. Chart of Accounts Materialization

hierarchies for processing (such as employees for payroll). We also found it necessary to program some data base triggers [Eswaran, 1976] to account for items such as an increase in quantity on hand for finished goods upon the completion of a factory job in process.

This discussion of account materialization concludes our treatment of the events accounting aspects of our data base design. In the next section of the paper, we will discuss the implications of this type of structure on DSS development, especially as it relates to the capabilities of a generalized problem processing system like KMAN.

4. DSS USE OF AN EVENTS MODEL

4.1. Introduction

Although there are certainly many different ways of looking at decision support systems, we will confine ourselves to the type of situation illustrated in Figure 6 where a decision maker is using an interactive computer facility to support judgement in an environment characterized by semistructured problems. These three DSS assumptions were the same ones made at the outset of the paper, and it is our intent at this point to discuss how such a system might work in an REA-modelled environment. As we go through these discussions, we will be pointing out implemented or planned features of our KMAN information system.

4.2. Critical Role of the Internal Data Base and the Need for an Events Approach to its Development

Simon's model of the decision-making process [Simon, 1960] consists of the three phases of intelligence, design, and choice. Although the flow of activities indicated by this model has been questioned by some, it does provide a useful means for identifying decision-making activities in managers. Textbook presentations (such as [Davis and Olson, 1985; Sprague and Carlson, 1982; Bonczek et al., 1981]) and other review discussions of DSS construction and evaluation [Lerch and Mantei, 1983] routinely demonstrate the general nature of computerized decision support by first of all listing example activities (such as

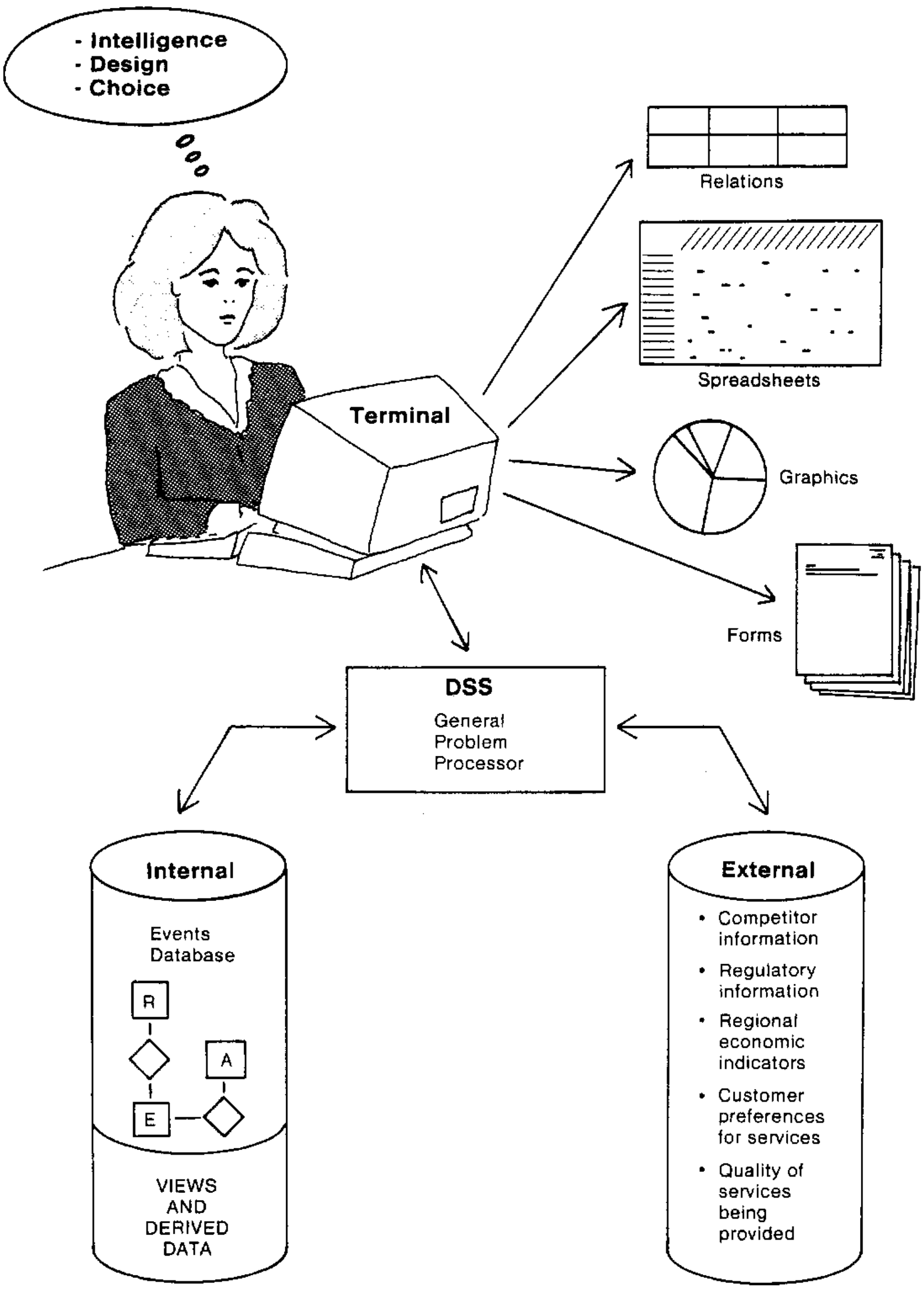

Figure 6. Events Accounting Foundation for DSS Environment

structured search and the generation of solutions) for each of these three phases and by then conjecturing on what type of computer facilities would be needed to support these activities. In all three phases, but especially in intelligence and design, the internal data base provides an essential starting point for a significant percentage of decision activity [Holsapple and Whinston, 1983]. These DSS presentations and discussions certainly point out that this support must be augmented by a data base of external information (including items such as those listed in Figure 6 [Davis and Olson, 1985, p. 44]) and by a collection of modelling techniques (such as statistical analysis and simulation). However, the necessity of these other components does not diminish the significance of a well-developed computerized store of past economic activity (such as sales, production, and financial transactions).

Although conventional accounting systems do capture data concerning economic phenomena which would be of interest to non-accounting decision makers, it is not uncommon to see discussions [Kotler, 1984; Shapiro and Kirpalani, 1984; Armitage, 1983] of these non-accounting decision activities mention the difficulties caused by accounting aggregation, classification, and allocation practices [American Accounting Association, 1969, 1971]. Suggested solutions to these problems can vary from (1) cultivation of alternative outside data sources to (2) complete reprocessing of source transactions to (3) minor manipulation of the final accounting numbers. Obviating the need for the first two of these alternatives (and the range of choices closely allied to them) is clearly the goal of a semantically-modelled events accounting system, and we have designed aspects of our KMAN prototype to show how we envision this simultaneous support of both accountants and non-accountants occurring.

In Figure 6, we have portrayed the internal data base as divided into an "events" part and a "views and derived data" part which actually correspond to what we called common data files and unique data files in Figure 1. The events data is represented as consisting of REA-modelled elements, but as we develop this prototype further, we hope to integrate this set of base tables with other base enterprise objects (such as an advertising campaign or a distribution channel). These other objects represent fundamental aspects of a corporate world to non-accounting decision makers, but they are most useful when they are portrayed in integrated fashion with related accounting entities such as sales or purchases. The views and derived data consist of elements such as our general ledger accounts which are predefined aggregations. The difference between views and derivations relates only to the periodicity of their update operations [McCarthy, 1982].

4.3. Example KMAN Operations

In Figure 6, a decision maker is portrayed engaging in the intelligence, design, and choice activities discussed earlier. Our KMAN prototype assumes again that this person is seeking support for these processes from an events-structured data base, and in that sense, our example DSS uses thus far have tended toward the "data-oriented" end of Alter's [1980] action implication spectrum. This is probably the most natural range of applications to be supported with an accounting data base, but as we proceed with development further, we expect to show some model-oriented use as well.

Also illustrated in Figure 6 is the variety of presentation methods KMAN allows for retrieval and use of relational data. These methods include presenting data in tables, presenting data as spreadsheets, presenting data in graphical terms (such as pie charts or scatter graphs), and presenting data within prespecified forms. As intimated previously, we have found this wide variety of presentation and extraction methods to be a very appealing feature of an integrated DSS support package. In our events implementation, we have tried to use our base of accounting data as a foundation upon which certain decision support activities could rely. We are, at present, still developing many support scenarios which place a hypothetical manager in decision-making settings similar to ones in which we would envision managers of our manufacturing object system finding themselves. The actual KMAN procedures and operations for some of the support scenarios which we have developed are illustrated in Appendix 4 of [Denna and McCarthy, 1985].

For presentation purposes in this present paper, we can outline some of our procedures and operations in abstract form for a DSS example used by Sprague and Carlson: the analysis of bad debts [Sprague and Carlson, 1982]. We assume that the decision maker in this case works in a company similar to that specified in our conceptual model (partially illustrated in Figure 4), that the decision maker has access to an events data base through an integrated DSS processor like KMAN, and that the decision maker is interested in a wide variety of customer and credit policies. More specifics concerning our examples in this case can be found in Buckley and Lightner [1973], Shapiro and Kirpalani [1984], and Kotler [1984].

1. Relational retrieval. If a manager was interested in viewing a list of customers with bad credit histories, he or she could be provided with such data in tabular form by having a relational processor (such as KMAN) join and select data from the tables representing customers, sales, sales returns and allowances, and cash receipts. A particular manager could apply any definition to the bad credit subset of customers, and that manager could also look at a wide variety of other customer and transaction attributes. Bad credit customers could also be subtyped by salesperson or product classes. Any manager could be given an individualized view of the disaggregate transaction data, and different concurrent views could be supported without one affecting the other. Tables such as these would be useful in both the intelligence and design phases of bad debt analysis.

2. Spreadsheet population and use. Similar to the instantiation of the derived tables above, disaggregate data could be used from an events data base to populate a spreadsheet according to the cell definitions of a particular manager. That particular manager could then experiment with different assumptions about credit-granting within a certain limit for overall bad debt expense. The rows and columns under such a scenario could represent items such as customers (grouped by credit risk category), aging categories for receivables

(derived from records of sales and cash receipts), or product groupings (derived from relationships linking sales to inventory). Again, such data would be most useful in the design stages of credit decisions.

3. Graphical output. Two graphical representations proposed by Sprague and Carlson [1982, p. 102] for use in the design and choice phases of bad debt analysis can be produced readily from an events data base with a DSS processor like KMAN. These representations are: (a) "A scatterplot of customers by two attributes associated with bad debts used to partition customers into risk groups" and (b) "A pie chart of percentage of loans by customer risk groups used to evaluate the partition." The customer attributes mentioned are surrogates for the customers' character and capacity to pay obligations, and these would be available as columns in the customer relation. The pie charts would actually track receivables (rather than loans) in our example enterprise, but the DSS principles would be the same. As mentioned above, receivables are obtainable from relations representing customers and the various economic events of customers.

5. CONCLUSION

Although we are only partially done with the construction of our events accounting prototype, we feel confident that the rest of this project will demonstrate clearly the benefits of an events basis for DSS construction and use. Knowledge Manager's abilities to perform a wide variety of aggregation and analysis tasks while simultaneously handling a full range of relational data base operations make it an excellent tool for our project, and we hope to enhance those capabilities even further with development of more complete and realistic cases.

In our future work, we intend to concentrate on the development of a general model base which will allow a decision maker to do extrapolation and forecasting on the basis of economic event sets. The computational methods for such generation techniques are straightforward, and it only remains to make their use simple for decision makers. We also would like to try to integrate our internal data bases with some outside data and to store the results in views for possible DSS use. And finally, we would like to build up our set of base tables further and to see if the ideas of events accounting can be more fully integrated with models and data from the areas of manufacturing, marketing, and distribution.

REFERENCES

Alter, S. (1980). Decision Support Systems: Current Practice and Continuing Challenges, Addison-Wesley.

American Accounting Association (1969). "Report of Committee on Managerial Decision Models," The Accounting Review (Supplement 1969), pp. 43-76.

American Accounting Association (1971). "Report of the Committee on Non-Financial Measures of Effectiveness," The Accounting Review (Supplement 1971), pp. 164-211.

Armitage, H.M. (1983). "Toward an Improved Distribution Accounting Information System Through an Entity-Relationship Modeling Approach," Ph.D. Dissertation, Michigan State University.

Atzeni, P., Batini, C., Lenzerini, M. and Villanelli, F. (1983). "INCOD: A System for Conceptual Design of Data and Transactions in the Entity-Relationship Model," in Entity-Relationship Approach to Information Modeling and Analysis, P. Chen (ed.), North-Holland, pp. 379-414.

Bonczek, R., Holsapple, C.W. and Whinston, A.B. (1981). Foundations of Decision Support Systems, Academic Press.

Borgida, A., Mylopoulos, J. and Wong, H.K.T. (1984). "Generalization/Specialization as a Basis for Software Specification," in On Conceptual Modelling, M. Brodie, J. Mylopoulos and J. Schmidt (eds.), Springer-Verlag, pp. 87-114.

Brodie, M.L., Mylopoulos, J. and Schmidt, J.W. (eds.) (1984). On Conceptual Modelling, Springer-Verlag.

Buckley, J.W. and Lightner, K.M. (1973). Accounting: An Information Systems Approach, Dickenson Publishing.

Chen, P.P. (1976). "The Entity-Relationship Model -- Toward a Unified View of Data," ACM Transactions on Database Systems, March, pp. 9-36.

Davis, G.B. and Olson, M. (1985). Management Information Systems, McGraw-Hill, Inc.

Date, C.J. (1981). An Introduction to Database Systems, 3rd ed., Addison-Wesley.

Demarco, T. (1979). Structured Analysis and System Specification, Prentice-Hall.

Denna, E.L. and McCarthy, W.E. (1985). "An Events Accounting Foundation for DSS Implementation," Working Paper, Department of Accounting, Michigan State University.

Eswaran, K.P. (1976). "Aspects of a Trigger Subsystem in an Integrated Database System," Proceedings of Second International Conference on Software Engineering, IEEE, pp. 243-250.

Financial Accounting Standards Board (1980). Statement of Financial Accounting Concepts No. 3: Elements of Financial Statements of Business Enterprises, FASB.

Gal, G. and McCarthy, W.E. (1983). "Declarative and Procedural Features of a CODASYL Accounting System," in Entity-Relationship Approach to Information Modeling and Analysis, P. Chen (ed.), North-Holland, pp. 197-213.

Gal, G. and McCarthy, W.E. (1985a). "Operation of a Relational Accounting System," Advances in Accounting, forthcoming.

Gal, G. and McCarthy, W.E. (1985b). "Specification of Internal Accounting Controls in a Database Environment," Computers and Security, March, pp. 23-32.

Gal, G. and McCarthy, W.E. (1985c). "An Artificial Intelligence Framework for Conceptual Specification and Evaluation of Internal Control," Paper presented at National TIMS-ORSA Meeting, April.

Hammer, M. and McLeod, D. (1981). "Database Description with SDM: A Semantic Database Model," ACM Transactions of Database Systems, September, pp. 351-386.

Holsapple, C.W. and Whinston, A.B. (eds.) (1983). Data Base Management: Theory and Applications, D. Reidel Publishing Company.

Holsapple, C.W. and Whinston, A.B. (1984). "Building Blocks for Decision Support Systems," Proceedings of the NYU Symposium: New Directions for Database Systems, New York, NY, pp. 121-164.

Howe, D.R. (1983). Data Analysis for Data Base Design, Edward Arnold.

Ijiri, Y. (1975). Theory of Accounting Measurement, American Accounting Association.

Keen, P.G.W. (1976). "'Interactive' Computer Systems for Managers: A Modest Proposal," Sloan Management Review, Fall.

Kent, W. (1978). Data and Reality, North-Holland.

Kotler, P. (1984). Marketing Management: Analysis, Planning and Control, Fifth Edition, Prentice-Hall.

Lerch, F.J. and Mantei, M. (1983). "A Framework for Computer Support in Managerial Decision Making," in Proceedings of the Fifth Conference on Information Systems, L. Maggi, J. King and K. Kraemer (eds.), ICIS, pp. 129-139.

Lum, V., Ghosh, S., Schkolnick, M., Jefferson, D., Su, S., Fry, J., Teorey, T. and Yao, B. (1979). "1978 New Orleans Data Base Design Workshop Report," Research Report RJ2554, IBM Research Laboratories, San Jose.

Martin, J. (1982). Strategic Data-Planning Methodologies, Prentice-Hall.

Martin, J. (1983). Managing the Data-Base Environment, Prentice-Hall.

Mattessich, R. (1964). Accounting and Analytical Methods, Richard D. Irwin.

McCarthy, W.E. (1979). "An Entity-Relationship View of Accounting Models," The Accounting Review, October, pp. 667-686.

McCarthy, W.E. (1981). "Multidimensional and Disaggregate Accounting Systems: A Review of the 'Events' Accounting Literature," MAS Communication, July, pp. 7-13.

McCarthy, W.E. (1982). "The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment," The Accounting Review, July, pp. 554-578.

McCarthy, W.E. (1984). "Materialization of Account Balances in the REA Accounting Model," Invited Presentation to the British Accounting Association, Norwich, England, April.

Shapiro, S. and Kirpalani, V. (1984). Marketing Effectiveness: Insights from Accounting and Finance, Allyn and Bacon, Inc.

Simon, H.A. (1960). The New Science of Management Decision, Harper and Brothers.

Smith, J.M. and Smith, D.C.P. (1977). "Database Abstractions: Aggregation and Generalization," ACM Transactions on Database Systems, June, pp. 105-133.

Sorter, G.H. (1969). "An 'Events' Approach to Basic Accounting Theory," The Accounting Review, January, pp. 12-19.

Sowa, J.F. (1984). Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley.

Sprague, R. and Carlson, E. (1982). Building Effective Decision Support Systems, Prentice-Hall.

Tsichritzis, D.C. and Lochovsky, F.H. (1982). Data Models, Prentice-Hall.

Winograd, T. (1975). "Frame Representations and the Declarative-Procedural Controversy," in Representation and Understanding, D.G. Bobrow and A. Collins (eds.), Academic Press.