The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment

William E. McCarthy

ABSTRACT: This paper proposes a generalized accounting framework designed to be used in a shared data environment where both accountants and non-accountants are interested in maintaining information about the same set of phenomena. This framework, called the REA accounting model, is developed using data modeling techniques, and its underlying structure is found to consist of sets representing economic resources, economic events, and economic agents plus relationships among those sets. Correspondence of REA elements with the accounting theories of Ijiri and Mattessich is discussed. Finally, practical use of the model in the database design phases of view modeling and view integration is presented, and some REA representations of accounting objects are reconciled with those representations found in conventional double-entry systems.

THE extension of the conventional accounting model to accommodate a broader spectrum of management information needs has become a topic of continued research interest since the 1960s. At that time, it became apparent that computerized data processing facilities would effect major changes in the way companies maintain their corporate stores of data, and some accountants perceived this transition period as an opportune time to rethink some of the basic constructs of traditional double-entry bookkeeping.

Among such accountants were two research committees of the American Accounting Association, one dealing with managerial decision models [AAA, 1969] and the other dealing with non-financial measures of effectiveness [AAA, 1971]. The restructuring analysis performed by these groups can be summar-ized by listing the following weaknesses that they identified in the conventional accounting model [McCarthy, 1980, p. 628].

(1) Its dimensions are limited. Most accounting measurements are expressed in monetary terms: a practice that precludes maintenance and use of productivity,

William E. McCarthy is Assistant Professor of Accounting, Michigan State University.

554

performance, reliability, and other multi-dimensional data.

(2) Its classification schemes are not always appropriate. The chart of accounts for a particular enterprise represents all of the categories into which information concerning economic affairs may be placed. This will often lead to data being left out or classified in a manner that hides its nature from non-accountants.

(3) Its aggregation level for stored information is too high. Accounting data is used by a wide variety of decision makers, each needing differing amounts of quantity, aggregation, and focus depending upon their personalities, decision styles, and conceptual structures. Therefore, information concerning economic events and objects should be kept in as elementary a form as possible to be aggregated by the eventual user.

(4) Its degree of integration with the other functional areas of an enterprise is too restricted. Information concerning the same set of phenomena will often be maintained separately by accountants and non-accountants, thus leading to inconsistency plus information gaps and overlaps.

Most of the research aimed at correcting the types of weaknesses mentioned above has concentrated on integrating Sorter's [1969] "events" accounting theories with *database approaches* to information management, approaches that assume that an enterprise chooses to manage its data as a centrally-controlled resource to be shared among a wide range of users with highly diverse needs. Accounting systems built with this type of orientation have included *hierarchical* models (Colantoni, Manes and Whinston [1971], Lieberman and Whinston [1975], Haseman and Whinston [1976]), *network* models (Haseman and Whinston [1977]), and *relational* models (Everest and Weber [1977]).

In 1979, a more generalized approach to the task of constructing events accounting systems in a database environment was proposed. Using the design methods of Chen [1976], McCarthy [1979] built an accounting system based on the primitive notions of entity and relationship sets and concluded with a data model or schema that could be mapped into any of the more specific approaches mentioned above. An important feature of this Entity-Relationship (E-R) work was its emphasis on the *semantic expressiveness* of the corporate data model: that is, on the degree to which elements in the final enterprise schema correspond to or capture the meaning of elements in the modeled corporate reality. The expressiveness of the earlier database models (hierarchical, network, relational) was limited, and this hampered their ability to specify the meaning of complex databases [Hammer and McLeod, 1981, p. 353].

This present paper extends McCarthy's approach. It explores the issue of database design in a larger organizational context, and it expands the E-R accounting methodology to include the concept of generalization hierarchies advanced by Smith and Smith [1977b]. Database abstraction is used to develop a generalized E-R representation of accounting phenomena that facilitates the conceptual modeling of enterprise-wide schemata. Recent literature in computer science [Brodie and Zilles, 1981] suggests strongly that information systems built with such conceptual modeling are better able to support multiple "views" of a centrally defined database. In an accounting context, the availability of multiple views would allow diverse and flexible use of economic transaction data, and it would provide a database designer with the capability to construct information systems free of the types of weaknesses identified earlier by the accounting research committees [AAA, 1969; 1971]. Each of those flaws can be corrected by orienting the accounting model-building

process toward a shared usage perspective.

The remainder of the paper is organized as follows. The first section reviews briefly the process of database design in a multi-user environment where both accountants and non-accountants have their information needs serviced via shared access to a centrally defined corporate store of data. An emphasis throughout this section is placed on the need for a conceptual schema: a schema which contains explicit definitions of the entities being modeled in the database and their properties and relationships [Tsichritzis and Klug, 1978, p. 180]. The second section of the paper develops a generalized E-R representation of accounting phenomena called the REA accounting model after its primary components which consist of sets representing economic *resources*, economic *events*, and economic *agents*. The REA framework is developed from an analysis of traditional account structures, and it is explained using the ideas of a number of accounting theorists, principally Yuji Ijiri [1975] and Richard Mattessich [1964]. Finally, the paper's third section discusses how REA representations of certain accounting phenomena relate to those representations used in traditional debit-credit systems. Additionally, this last section identifies specific situations where modifications to the model's basic structure might be appropriate.

The topic of database design follows next. This section provides the reader with a review of logical design concepts and establishes the importance of an entity-based description of an enterprise's database.

DATABASE DESIGN AND THE CONCEPTUAL SCHEMA

A database can be defined "as the model of an evolving physical world" [Abrial, 1974, p. 3]. Database design,

therefore, is a process during which an attempt is made to mirror aspects of an identified reality (called the *object system*) in an abstract model (called the *data model* or *schema*). For implementation purposes in an actual enterprise, this data model or schema is then translated into the definition language of a database management system (DBMS) and operated in a specific software and hardware environment.

For simple object systems with little organizational complexity, schema design involves a relatively straightforward modeling process, because potential users are of one mind in delineating the "things" of interest to be contained in the database. An example of such a simple system is the small retail enterprise modeled by McCarthy [1979]. For more complex object systems however, the design process must account for the difficulties associated with identifying and integrating the information needs of many different users. To handle this increased level of complexity, Lum et al. [1979, p. 108] suggest that data modeling be separated into the following phases: (1) requirements analysis, (2) view modeling, and (3) view integration. Each of these three steps is explained and then summarized in the subsection below.

Data Model Development

Requirements analysis is a process during which analysts interview the user community and review existing documentation in an effort to identify both present and future information needs.[1] For purposes of developing a data model,

[1] There is an extensive body of computer science and information systems literature that deals with the topic of requirements analysis in considerably more detailed fashion than that specified here. Recent overviews of this literature have been produced by Cooper and Swanson [1979] and Taggart and Tharp [1977], and some particular analysis methodologies are outlined in De-Marco [1979] and Munro and Wheeler [1980].

three items need to be specified during this design phase:

(1) information concerning the processes (and decisions) that use data;
(2) information concerning the various data elements themselves and their patterns of usage across processes; and
(3) information concerning the various organizational constraints on data use.

In simpler terms, the purpose of requirements analysis is to summarize individual answers to the following question as it is posed throughout the organization: "Which items of data are needed either to complete this process or to make this decision?" Each data list given in reply to this query constitutes a local *view* or individual user perspective of the entire database. The purpose of the next two design phases is to combine these local views into an enterprise-wide (or global) view.

View modeling is a process during which the database designers take each of the local views gathered during requirements analysis and prepare them for integration by characterizing the view's individual components in terms of a *semantic* data model, such as Chen's [1978, p. 30] E-R framework. These semantic models organize data elements around the primitive notions of entities, relationships, and attributes, and it is the designer's task at this stage to identify those basic realities as they exist in the object system. Frequently, this identification requires either a high level of expertise with the functional areas of an organization (such as accounting, marketing, personnel, etc.) or a high level of interaction and feedback with the original data users. It is at this point in the design process that this paper proposes to use the REA framework to identify entity-based descriptions of accounting phenomena.

View integration completes the development of a data model. During this phase, the designers first combine the local views (which are expressed in semantic terms) into a global data model. Then, they specify how each of those local views can be derived from that combined framework. The REA model should also provide design assistance for these steps.

In summary, the output of the data model design process[2] should consist of two items: (1) an enterprise-wide model of data (global view), and (2) a set of local views accompanied by procedural specifications for their derivation from the global view. The common terms for these global and local views respectively are *conceptual schema* and *external schema* [Tsichritzis and Klug, 1978]. A third type of specification—the *internal schema* —is also important to database development, but because internal schemata concern physical storage considerations, they will not be explained in detail here.

The Conceptual Schema and Semantic Descriptions

The preceding review of database design phases should indicate to the reader the importance of both the conceptual schema and the semantic vehicle used to express its elements. Because of this importance, a number of researchers have developed their own semantic frameworks in recent years and proposed them for use in database design. Such frameworks are now termed "second generation data models" [Lum et al., 1979, p. 32], and they include among

---

[2] Data model development is not an aspect of database design unique to the methodology proposed by Lum et al. [1979]. The development phases outlined here are common to many design processes. In most cases however, the phases are given slightly different names. For examples, see Teorey and Fry [1980], Kahn [1978], and Flavin [1981].

their number those models advanced by Chen [1976], Smith and Smith [1977a; 1977b], Bachman and Daya [1977], Codd [1979], Mylopoulos, Bernstein, and Wong [1980], and Hammer and McLeod [1981].

Though all of these second generation frameworks differ in the specific terms they use and in the detailed definitions assigned to those terms, they do use certain semantic notions that are essentially similar [Lum et al., p. 34]. Each of the models supports the notion of an *entity* as a person, object, or happening that is modeled in the database. Additionally, they all support two types of *relationships*. The first type of relationship has already been described in the accounting literature by McCarthy [1979], and what it does is relate information objects (or entities) of different types to form new objects. For example, the relationship "pays for" is formed by connecting the entities "cash receipt" and "sale," or the relationship "participates in" is formed by connecting the entities "vendor" and "purchase." Such relationships are called *associations*. The second type of relationship supported by second generation models has not yet been presented in an accounting context and thus will be explained in more detail below and illustrated using Figure 1.

The second relationship type is called *generalization*, and its definition is due primarily to Smith and Smith [1977b]. What generalization does is to relate different subtypes or subsets of entities to a generalized type or superset. For example, the entities "raw material," "work in process," and "finished goods" generalize to the entity "inventory."

Some instances of generalization are shown in Figure 1(a). At each level in the hierarchy, the entities shown in the boxes generalize to the next higher level. Thus "clerk," "salesperson," and "profes-

sional" all generalize to "employee," or alternatively "clerk" is a category or subset of "employee," "customer" is a category or subset of "economic agent," etc. In Figure 1(b), generalization is illustrated in a different way: as an operation occurring out of the page in simulated three-dimensional space. This convention was adopted by Smith and Smith [1977b], and it will be used in later sections of the paper.
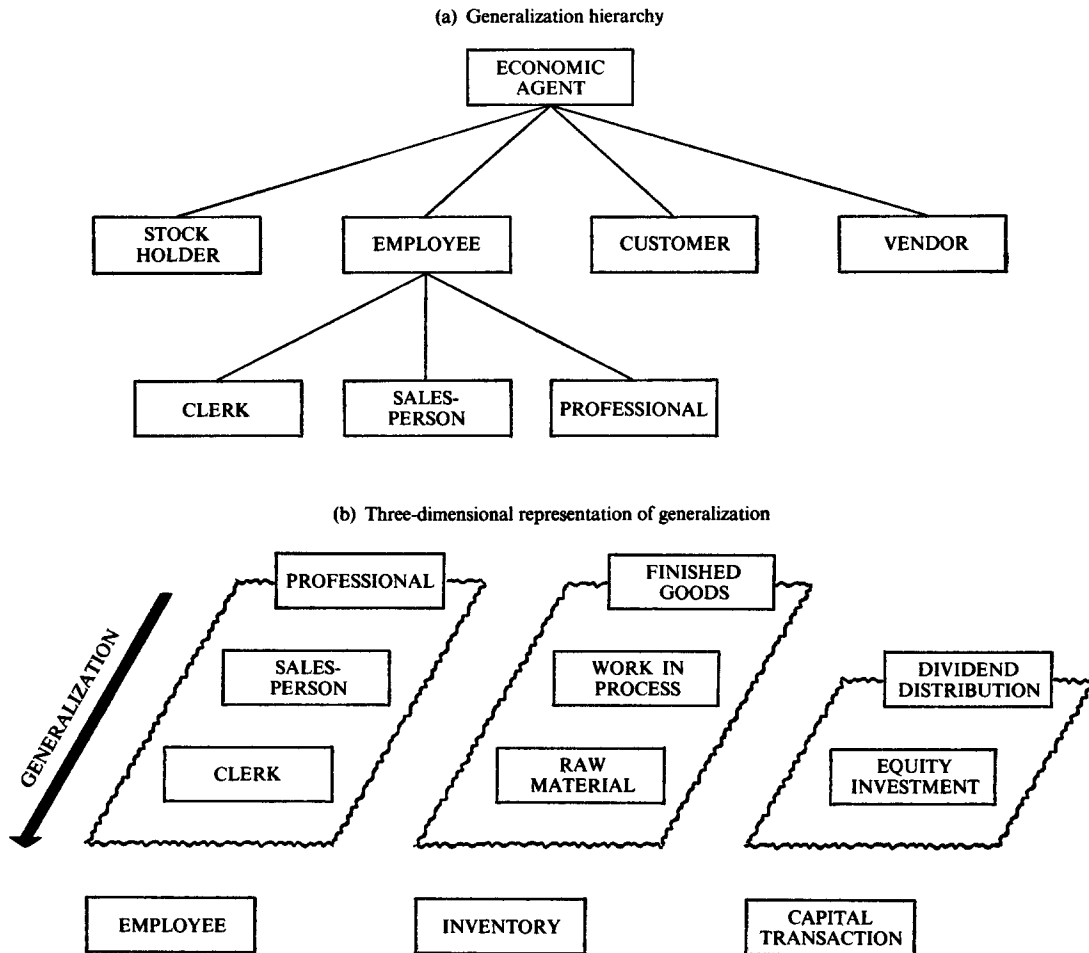
In Figure 2, an overall example of database design with second generation concepts is illustrated. The focal point of the . database is a conceptual schema which characterizes the object system in terms of entities (boxes), association relationships (diamonds), and generalization relationships (three-dimensional operations). Shown on the right of the figure are three local views of data needed to satisfy these example users:

(1) a customer service representative responsible for questions concerning open orders;
(2) a general ledger clerk responsible for preparing trial balances; and
(3) an inventory clerk responsible for controlling raw materials.

As indicated by the dotted lines, an actual database implementation would consist of a conceptual schema with many more information objects and a much larger set of external schemata.

To summarize the data model design process in terms of Figure 2, it can be said that *requirements analysis* consists of specifying the local views of data or external schemata, that *view modeling* consists of restating these views in terms of a semantic model, and finally that *view integration* consists of combining the semantically-expressed local views into a global view and determining procedures for materializing the external schemata from that combined framework.

FIGURE 1
GENERALIZATION EXAMPLES

(a) Generalization hierarchy

```
                        ┌──────────┐
                        │ ECONOMIC │
                        │  AGENT   │
                        └──────────┘
            ┌─────────────┬──┴──────┬──────────────┐
       ┌────────┐    ┌──────────┐ ┌──────────┐  ┌────────┐
       │ STOCK  │    │ EMPLOYEE │ │ CUSTOMER │  │ VENDOR │
       │ HOLDER │    └──────────┘ └──────────┘  └────────┘
       └────────┘   ┌──────┼──────┐
              ┌───────┐ ┌────────┐ ┌──────────────┐
              │ CLERK │ │ SALES- │ │ PROFESSIONAL │
              └───────┘ │ PERSON │ └──────────────┘
                        └────────┘
```

(b) Three-dimensional representation of generalization

This summary concludes the review of database design processes. The next section features development of a tool to be used during these processes—the REA accounting model.

THE REA ACCOUNTING MODEL

The reader may now appreciate the change in perspective needed if accounting is to move away from the position of being an independent and non-integrated information system (a position subject to the criticisms outlined at the beginning of the paper) and toward the position of being a constituent part of an enterprise database system. For such movement to occur, it will be necessary to have accounting phenomena characterized in terms compatible with the design phases of view modeling and view integration. The REA framework developed in this section is such a characterization.

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

FIGURE 2

SCHEMA SPECIFICATION FOR A DATABASE
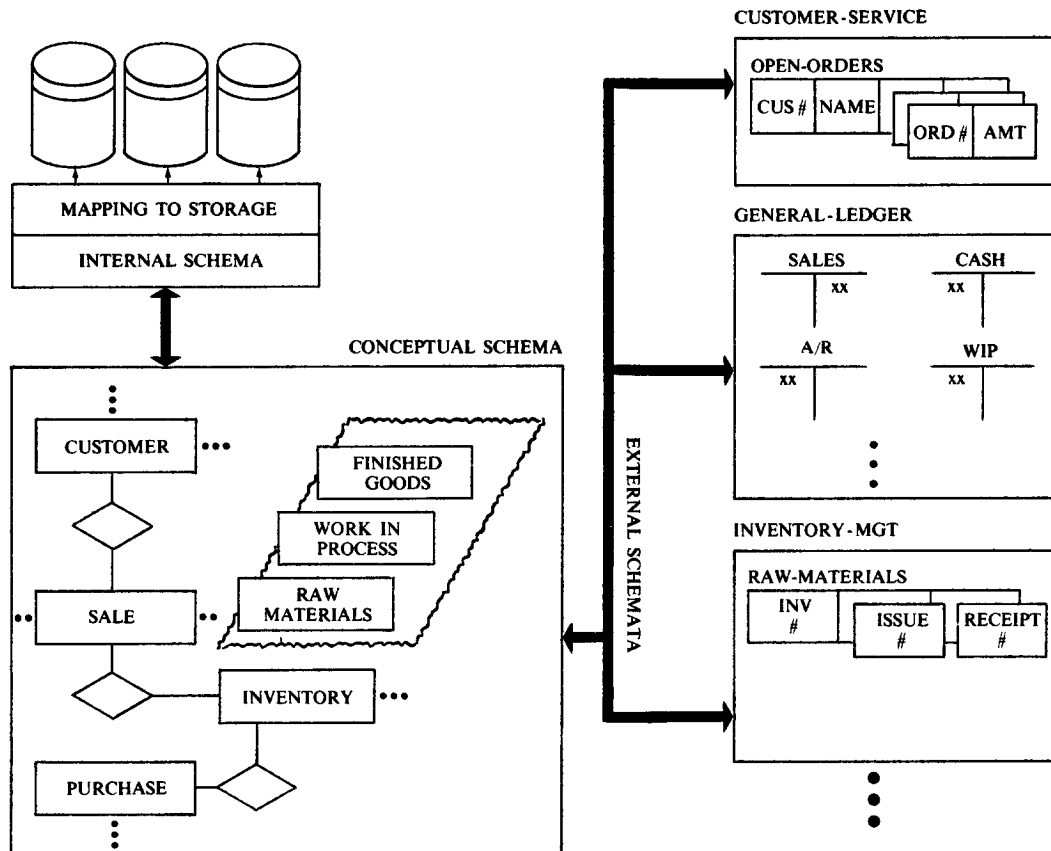
as debits, credits, and accounts. As noted previously by both Everest and Weber [1977] and McCarthy [1979], these elements are artifacts associated with journals and ledgers (that is, they are simply mechanisms for manually storing and transmitting data). As such, they are not essential aspects of an accounting system. It is possible to capture the essence of what accountants do and what things they account for by modeling economic phenomena directly in the conceptual schema. Any double-entry manipulations desired by particular users can then be effected only in the external schemata presented to those users.

What this paper proposes to use instead of bookkeeping schemes is an accounting framework whose structure is derived with semantic modeling and whose elements correspond closely to principles expressed by accounting theorists such as Ijiri [1975], Mattessich [1964], Buckley and Lightner [1973], and Yu [1976]. Development and explanation of this framework will begin with a consideration of economic resources and economic events—the two entity classes which constitute the subject matter for balance sheets and income statements— and proceed with an analysis of economic agents and a subset of economic agents:

FIGURE 3

GENERALIZATION OF ACCOUNTING STOCKS AND FLOWS
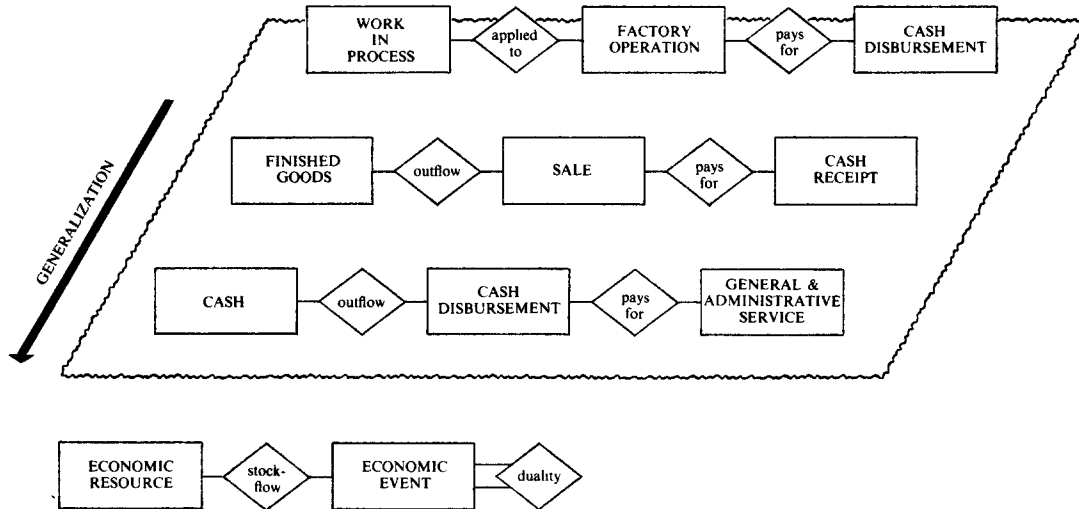
economic. units. Practical application of this entire REA structure is discussed in the last section of the paper.

Economic Resources and Events

In examining the aspects of an enterprise that are of interest to accountants, the basic stock-flow nature of accounting information becomes readily apparent. Elements of the general ledger normally are classified as either balance sheet accounts, which represent monetary *stocks* of goods, services, and claims at a particular time, or income statement accounts, which represent monetary *flows* of these same items over a period of time. This basic dichotomy points correctly toward classification of accounting phenomena into two generalized categories: stock objects and flow transactions. However, for data modeling purposes, the composition of these categories must be altered somewhat.

A chart of accounts cannot be mapped directly to an accounting data model because many of its elements represent procedural aspects of the double-entry

recording process. In abstracting from accounting practice to a generalized schema, it becomes necessary to concentrate on the entities to be accounted for (and their relationships and attributes) rather than on the classification and reporting methods to be used. 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," and they therefore should be clustered together [Chen, 1976]. Further problems with using procedural and classificational frameworks as models are noted by Everest and Weber [1977].

Characterization of accounting phenomena in terms of resources (objects), events, and the association between such entities was carried out by McCarthy [1979, pp. 671–75]. Similar representations are shown in Figure 3 and then generalized to an overall structure that reflects the stock-flow aspects of accounting object systems. The nature of each of the generalized components is explained below.

Economic resources are defined by Ijiri [1975, pp. 51–2] to be objects that (1) are scarce and have utility and (2) are under the control of an enterprise. In practice, the definition of this entity set can be considered equivalent to that given the term "asset" by the FASB [1979, pp. 51–7] with one exception: economic resources in the schema do not automatically include claims such as accounts-receivable. This exception will be clarified later in the paper.

Economic events are defined by Yu [1976, p. 256] as "a class of phenomena which reflect changes in scarce means [economic resources] resulting from production, exchange, consumption, and distribution." For both Ijiri [1975] and Sorter [1969], economic events constitute the critical information elements of an accounting system; information concerning the resource set is viewed only as an indirect communication of the full event history. When the model illustrated in Figure 3 is considered in its maximum form of temporal generality, (without considering implementation cost), event descriptions would be maintained perpetually as *base elements* of the conceptual schema. That is, detailed descriptions of all transactions would be stored indefinitely in disaggregated, individual form. In practice, this is an unrealistic assumption, and variations will be discussed when the subject of REA modifications is addressed.

Stock-flow relationships simply connect the appropriate elements in the entity sets defined above. Again considering the model in terms of its maximum generality, a perfectly consistent schema would require both a new instance of this relationship type and a new update or instance of a resource entity type for every new event entity. As noted by Wong and Mylopoulos [1977, pp. 368–77], model-

ing such aspects of a changing physical world is best done by *procedure invocation* (such as the triggered updates used by McCarthy [1979, p. 682; 1980, p. 632]). However, large sets of procedures do not normally have to be written to account for stock-flow interplays because maintenance of less than perfect consistency is usually acceptable. This topic will also be discussed later in the paper.

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].

The generalized sets defined above describe the accounting universe in terms of the resources and events to be accounted for. An equally important part of that universe concerns the *participants* in the accounting process. In the next subsection, the paper will discuss the activities of participants both inside and outside of the enterprise.

Economic Agents and Units

As reflected in the classification schemes of a general ledger, the roles of participants in the economic affairs of an enterprise are accounted for in dual fashion. First, in a dynamic manner that involves parties both inside and outside of the company, specific *participation* in economic events is recorded. This application is reflected in the use of organizational unit codes for many expense and asset accounts and in the use of subsidiary ledgers for receivables and payables. Second, in a more static manner that involves only inside parties, *responsibility*

FIGURE 4

GENERALIZATION OF ACCOUNTING PARTICIPANTS
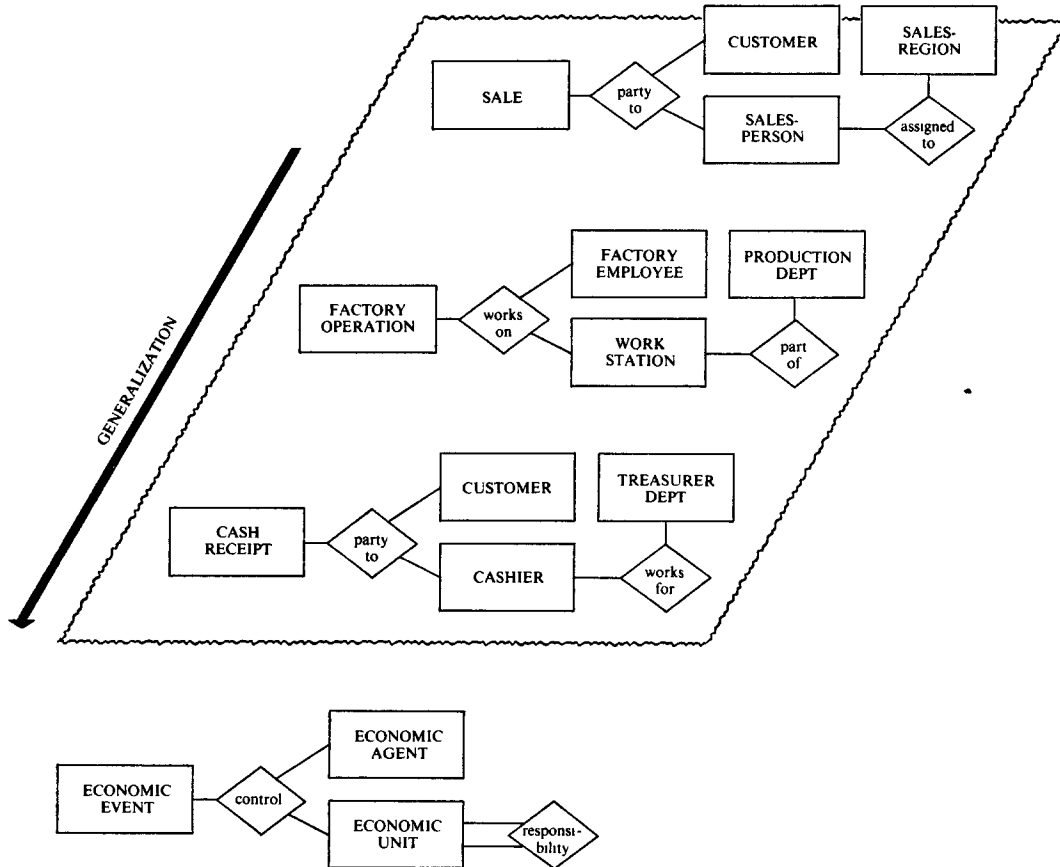
for the economic actions of subordinates is recorded. This application is reflected by incorporation of organizational responsibility charts into the coding of accounts.

Information concerning participants and their roles for the economic events "{sale, factory operation, cash receipt}" is shown aggregated to E-R form in Figure 4. The sets derived by generalizing these instances are explained below.

Economic agents include persons and agencies who participate in the economic events of the enterprise or who are re-

sponsible for subordinates' participation. Agents in this sense can be considered equivalent to what Ijiri [1975, pp. 51-2] calls "entities." That is, they are identifiable parties with discretionary power to use or dispose of economic resources.

Economic units constitute a subset of economic agents. Units are inside participants: agents who work for or are part of the enterprise being accounted for.

Control relationships are 3-way associations among (1) a resource increment/decrement (event), (2) an inside party (unit), and (3) an outside party (agent).

FIGURE 5

THE REA ACCOUNTING MODEL

(a) Entities and relationships
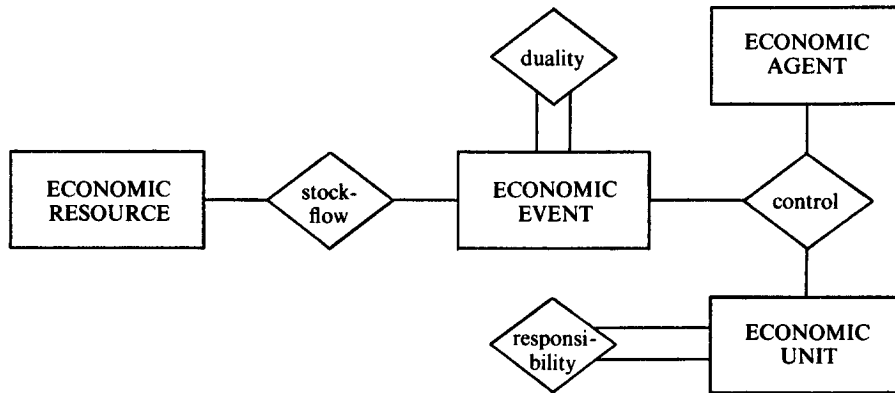
(b) Role declarations

stock-flow

Participating Entity Sets	ECONOMIC RESOURCE	ECONOMIC EVENT
Roles	stock	flow

duality

ECONOMIC EVENT	ECONOMIC EVENT
increment	decrement

responsibility

ECONOMIC UNIT	ECONOMIC UNIT
superior	subordinate

control

ECONOMIC EVENT	ECONOMIC AGENT	ECONOMIC UNIT
exchange transaction	outside party	inside party

The requirements underlying this relationship are best explained by Ijiri [1975, p. 52].

In general, an entity's power to control resources is provided by someone else, who in return demands that the entity account for the resources under its control. Therefore, accountability . . . and control . . . may be regarded as two sides of the same coin.

Responsibility relationships indicate that higher level units control and are accountable for the activities of subordinates. Because employees are considered economic units (controlling at a minimum their own services), this relationship set should include the hierarchical ordering of superior-subordinate agencies *and* the assignment of employees to those agencies. Manager assignment can be considered a category of employee assignment.

In summary, the role of participants in

the generalized model is best understood in terms of accountability and control. As Ijiri [1975, p. 52] notes, there are circumstances where these principles may not go hand-in-hand, but at least the model as constructed allows accountants to identify all aspects of both notions that originate with exchange transactions.

Generalized Framework Summary

When all the generalized elements of Figures 3 and 4 are combined, the E-R framework shown in Figure 5(a) results. In Figure 5(b), role declarations are illustrated for each of the generalized associations. The detailed use of these role declarations is explained by McCarthy [1979, pp. 679–81] and Kent [1978, pp. 63–5]. Their primary purpose is to specify which entity classifications are allowed to take part in an instance of an association. Additionally, however, they supply semantic information useful in differentiating relationship participants similar to each other such as superiors/subordinates and inside/outside parties.

The only static object in the generalized schema concerns responsibility; all other elements revolve around the dynamic representation and maintenance of information concerning economic events and how they relate to economic resources and economic agents (economic units are a subset of agents). In the next section of the paper, the tradeoff and implementation decisions involved in using this REA model as an operational database design tool are illustrated and analyzed in some detail.

DATABASE DESIGN WITH THE REA FRAMEWORK

In the previous section, a generalized E-R schema of accounting phenomena was developed for the express purpose of facilitating the view modeling and view integration phases of database design. This section will describe those design processes and outline some of the decisions and tradeoffs that might be appropriate in building an information structure acceptable to both accountants and non-accountants in a shared data environment. The analysis process envisioned here is one in which the database designer and accountant work together in (1) identifying the data requirements of different processes and decisions, (2) restructuring those data requirements in semantic terms using the REA model as an "instance framework," and (3) combining the various restructured data specifications to form a conceptual schema.

The discussion of design decisions and tradeoffs to be given in this section will be very specific and will relate to particular aspects of accounting practice and convention. To begin this exposition, the paper will use a simple illustration which concerns inventory purchases and cash payments.

Initial REA Example

The list of data elements concerned with processing of purchase transactions in an information system could include names like the following: purchase invoice number, vendor name, buyer employee number, stock numbers of line items, dollar amount of purchase, quantity and unit price of line items, etc. If such a list were identified during the requirements analysis stage of design, development of a data model would continue by trying to characterize those items in semantic terms (that is, in terms of entities and relationships). With the REA model of Figure 5(a) as a framework, this characterization process would proceed as follows:

(1) the economic event involved (pur-

FIGURE 6

INITIAL REA EXAMPLE
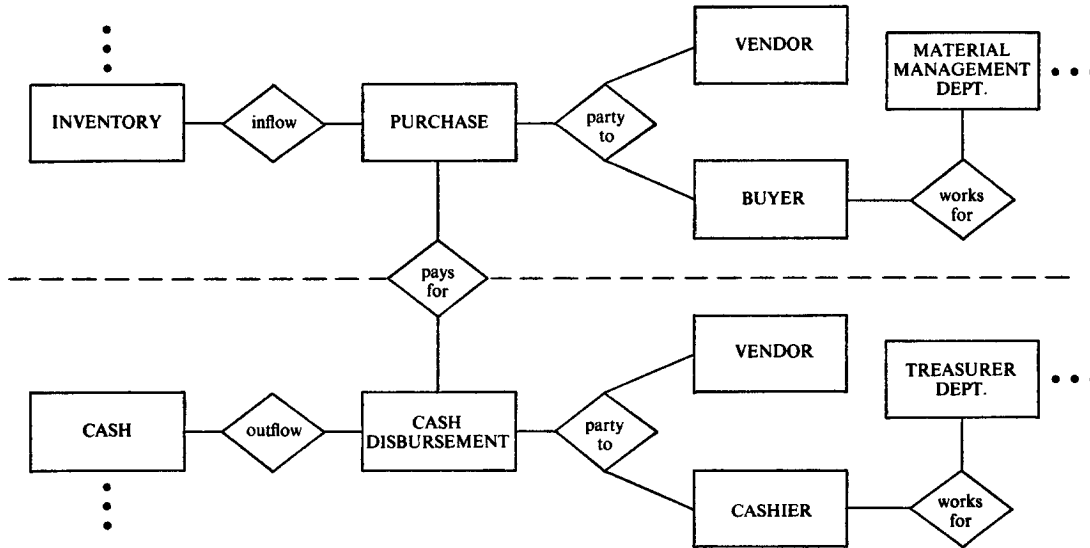
chases) would be identified, and data elements describing that event would be specified as event attributes;

(2) the economic resource that the event affected (inventory) would be identified along with its attributes; and

(3) the inside and outside participants in the event (buyers and vendors) would be identified along with their attributes and higher level economic units.

An E-R diagram illustrating the REA view modeling of "purchases" is shown in the upper part (above the dotted line) of Figure 6. Designation of attributes is not illustrated there, but such designation would follow methods outlined by Chen [1978]. In the lower part of Figure 6, view modeling is outlined for the separate requirement of cash disbursements processing.

The design phase of view integration involves combining local views into an enterprise-wide conceptual schema. The mechanics of this process are too complex to be explained completely at this point,[3] but the reader may gain an understanding of their nature by following the logic of the combination decisions discussed below for the "purchase" and "cash disbursement" example of Figure 6.

(1) First, the two transactions would be associated with a "pays for" relationship, because the REA framework specifies that all economic events be linked with *duality* associations. In this particular case, "purchase" fills the *increment* role, and "cash disbursement" the *decrement* role.

(2) Second, the two "vendor" sets would be combined, and inconsistent or overlapping attributes would be resolved.

[3] Readers interested in following a more detailed example of view integration may consult Teorey and Fry [1980, pp. 192–97].

(3) Third, some generalization decisions would have to be made. For example, it might be decided to decompose "inventory" into "{raw material, work in process, finished goods}" or to combine "{cashier, buyer}" into "employee." In some cases, it will make sense to designate both the generalized set and its subsets as information objects; in other cases, it will not. The logic for such decisions is explained by Smith and Smith [1978] and Smith [1978].[4]

(4) Finally, some clues could be discerned that would provide directions for other combinations. For example, an "outflow" association would be needed for "inventory" (linking in this case to processing of sale transactions) and an "inflow" would be needed for "cash" (linked to cash receipts processing). In a similar manner, the places of "treasurer dept." and "material management dept." in the organizational hierarchy would indicate other combination decisions.

As various accounting data requirements are gathered from different parts of a company, they would be combined with non-accounting decision needs to build the conceptual schema. An important priority of this combination process is ensuring that all phenomena in the object system are modeled in as consistent and non-redundant a fashion as possible. Thus, when other requirements are identified that relate to the resources, events, and agents illustrated in Figure 6, they would be integrated systematically rather than being modeled separately. Such requirements could include, for example, these two: (1) purchase order transactions to be associated with "purchase" and "vendor" in an inventory management plan and (2) various personnel training and benefit programs to be associated with employees like "cashiers" and "buyers."

The paper's exposition of an initial REA example is now complete. The rest of this database design section addresses two important issues, one of them very general and the other very specific. The specific issue involves a discussion of individual cases where REA modeling does not seem to fit precisely. However, to understand the problems surrounding these individual cases, it is necessary to discuss the general issue of *conclusion materialization* [Bubenko, 1977] first.

Conclusion Materialization

Simply stated, the process of conclusion materialization involves producing information "snapshots" from records of continuing activities. In an events accounting system, all information is derived from the events themselves, and an important consideration therefore is how to propagate and organize the data derived from transaction recording. For the REA model, two aspects of this question warrant discussion: (1) what things to materialize conclusions about, and (2) what methods to use in doing this. Each of these topics is discussed in a subsection below.

Things to Materialize Conclusions About: Resources and Claims

The process of producing information snapshots concerning accounting objects will be illustrated using the example given in Figure 7. In the system shown, certain attributes of the two resource entity sets ("inventory" and "cash")

---

[4] As mentioned previously, most data modeling frameworks explicitly support the specification of generalization relationships. Readers interested in seeing how such specification can be accomplished for an E-R data model should consult Schiffner and Scheuermann [1979], Hawryszkiewycz [1980], or Sakai [1981].

FIGURE 7

EVENT IMBALANCES AS RESOURCES AND CLAIMS
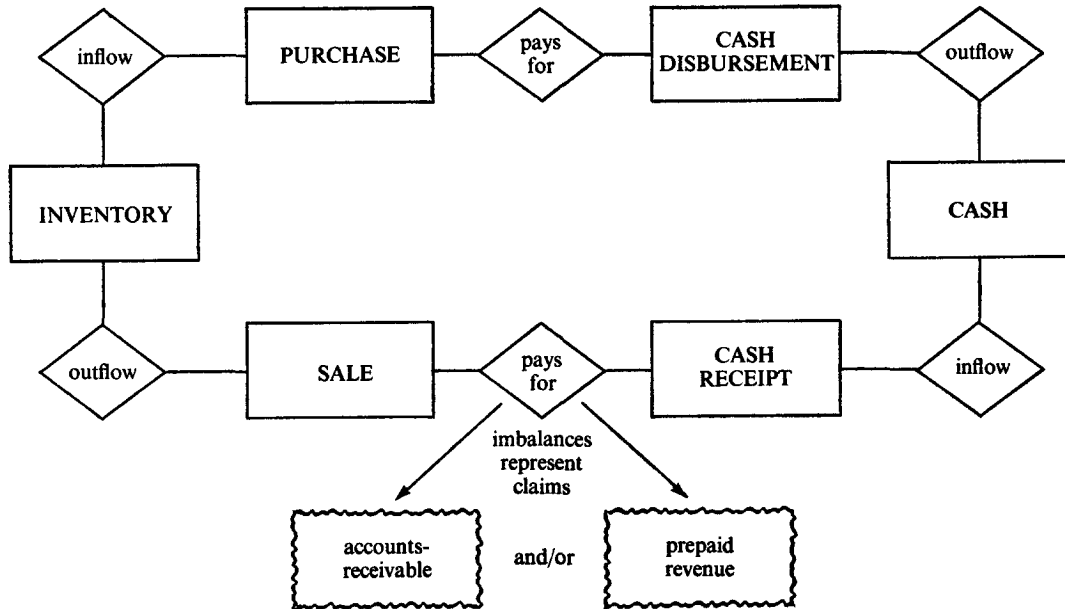
actually represent imbalances between their inflow and outflow events sets. For instance, the quantity on hand for a particular inventory item derives from an excess of purchases over sales for that item. If a designer required that all such stock characteristics of resources were to be updated immediately upon the occurrence of flow events, the data model would be *perfectly consistent*. In most circumstances, such perfection is not always needed. Therefore, what the designer must do during conceptual modeling is assess the decision usefulness and applications requirements of a certain consistency level, so that these benefits may be contrasted with implementation costs during later design phases.

In contrast to the "cash" and "inventory" examples shown in Figure 7 where imbalances between inflows and outflows are represented by characteristics of resource sets, the imbalances that

exist between "sales" and "cash receipts" and between "purchases" and "cash disbursements" represent not tangible resources but *claims* for and against the enterprise. Claims, or future assets as they are called by Ijiri [1975, pp. 66–68], derive from imbalances in duality relationships where an enterprise has either:

(1) gained control of a resource and is now accountable for a future decrement (future negative asset) or

(2) relinquished control of a resource and is now entitled to a future increment (future positive asset).

Cases concerning individual representation of claims are discussed later in the paper.

Methods for Producing Conclusions: Procedures

The REA accounting framework provides a basis for what can be called the *declarative* features of a conceptual

FIGURE 8

PROCEDURE TYPES

effect on base objects

		effect	*no effect*
consistency maintenance	*instant*	triggered procedure	view procedure
	periodic	adjustment procedure	derivation procedure

schema. These features model facts about the object system in terms of *entities*, *associations* of entities, and *generalizations* of entities. However, in a working data model, they must be supplemented by *procedures* that specify how the system will use those facts it has available [Wong and Mylopoulos, 1977]. In terms of concepts discussed already, the declarative features of an accounting schema consist of its *base objects*—those elements representing economic events, resources, and agents plus relationships between them —while the procedural features consist of methods for materializing conclusions about those base objects.

There are two particular aspects of procedural representation that must be understood when discussing the decisions and tradeoffs involved in data modeling. The first aspect concerns timing and involves a consideration of whether to use new data to update the database instantly or to use it to update at only periodic intervals. The second aspect concerns the effect of a procedure on the base objects and involves a consideration of whether to view certain data objects as fundamental characteristics of the reality being modeled or to view them as derived information of interest only to a small segment of the user community. The types of procedures concerned with these two aspects are illustrated in Figure 8 and explained below.

(1) Triggered procedures [Eswaran, 1976] are invoked when economic events are recognized and cause immediate updates to base objects. An example would be an update to an inventory item triggered by a sale or purchase.

(2) Adjustment procedures also affect base objects but only at specified intervals. An example would be a daily or weekly adjustment to a cash account done by adding receipts and subtracting disbursements.

(3) View procedures produce "dynamic windows on the data base" [Chamberlin et al., 1976, p. 566] through which different classes of users may separately view information. The conclusion or data elements materialized as a result of a view procedure are "virtual" in the sense that they exist only when the procedure is invoked. An example would be a view that allows accountants to look at imbalances between sales and cash receipts as accounts-receivable while simultaneously allowing other users to look at the same objects as detailed transaction histories.

(4) Derivation procedures produce—at periodic intervals—information derived from but not directly affecting base objects. Examples would be total advertising expenditures for a month or net income for a year.

The capability to support all four of these procedure types will vary among different database management systems, especially with regard to triggers and views. This is a matter that a database designer would have to account for during analysis and specification of both conceptual schema and external schema elements. Readers interested in a more extensive treatment of procedure definition and use may consult Date [1981].

Conclusion Materialization Summary

The paragraphs above have explained conclusion materializations in terms of the data objects they produce and the methods for effecting them. In discussing, under the heading of periodicity, this entire issue of producing point-in-time information for accounting purposes, Yu [1976, p. 262] has noted the serious difficulties in theory and practice that can arise:

> Periodicity forces more or less arbitrary division of economic activities between specific time intervals and hence gives rise to some of the most difficult issues in accounting, such as periodic allocation of expired resources, matching charges against revenues, the all-inclusive income concept, cash and accrual bases of accounting, estimates and approximations of expenses and revenues, determination of unexpired resources, and consistency and comparability of accounting information from period to period.

The REA framework and its accompanying procedural capabilities do not solve these problems. In most cases, the same difficult choices will have to be made if the system is to reflect current practice. What a data modeling process does offer,

however, is the ability to minimize the effect that arbitrary choices might have on other information users.

The paper will move now to a discussion of individual design decisions involved in adapting elements of the REA framework to specific aspects of accounting convention and practice.

Design Decisions Based on Specific Aspects of Existing Practice

As illustrated by the design example dealing with purchase and cash disbursement processing, the sets of entities and relationships outlined in Figure 5(a) constitute a framework for operational analysis of an enterprise's information needs. For example, if something is identified as an economic resource during view modeling, a database designer should expect that two event sets, one an inflow and the other an outflow, also will be specified. Furthermore, each of these two event sets would require inside and outside participants that would be linked via control relationships, and additional events that would be linked via duality relationships, and so on. Using this method of analysis, the designer and accountant could work together in specifying a conceptual schema that would range across the different functional areas of an enterprise and that would be able to accommodate both financial reporting and managerial control needs.

For some elements of accounting, it may appear that the REA framework does not apply. There are, for example, some accounting transactions that do not seem to affect any identifiable resource or do not seem to be linked causally with other transactions. It turns out, however, that such situations are not really anomalous. They only reflect instances where existing accounting convention allows less than full specification of schema elements. In the subsections following,

some of these situations are explored in detail, and REA representations of them are reconciled with double-entry representations. The paper also discusses here procedural implementations and situations where modification to the generalized framework might be appropriate.

The specific aspects of accounting data modeling that will be examined next in the context of existing practice are:

(1) treatment of imbalances in duality relationships (claims) as base objects;

(2) summation of economic event data over time;

(3) partitioning and combination of economic events;

(4) treatment of duality relationships at the macro-level; and

(5) equity transactions.

Claims as Base Objects

In the discussion surrounding Figure 7, it was noted that both resources and claims derived from imbalances in related event sets. However, in the outline of the generalized framework, resources were materialized as base objects while claims were not. In actual practice, this disparity in treatment may not always be warranted, especially when the processing requirements and decision usefulness of some claims are projected. The judgements involved in this issue are discussed below.

Claims are of interest in two ways:

(1) as independent data objects where they can have attributes of their own and can be categorized on subtypes of those attributes (for example, accounts-receivable classified by aging period or bonds classified by type of security);

(2) as attributes of economic agents where they find their primary usefulness in relating to the specific outside parties with whom future increment or decrement exchanges are expected (for example, accounts-receivable classified by customer or bonds classified by debtor and creditor).

Primacy of the first use described above justifies maintenance of claim data either as separate base objects, or as separate derived objects, or as views; primacy of the second use justifies only maintenance as separate characteristics of preexisting base objects. Compromises that range between these two alternatives probably will be common and will be influenced further by the need to materialize conclusions about the claims either immediately or periodically. Should the accountant and database designer decide together to maintain certain claims as separate base objects, they also would have to include two additional events sets (inflow and outflow) for each one.

The type of analysis applied to claims and outside parties above will occasionally apply to resources and inside parties in a somewhat parallel manner. That is, there will be times when the decision usefulness of resource information will be enhanced by maintaining a direct relationship between the resource concerned and the economic unit that controls it. A good example of this would be maintenance of quantity on hand data for different products stored in different warehouses. One accounting convention that would be especially well served by such an arrangement would be process costing, a procedure during which most events are tracked only by economic unit and for which resource conclusions are materialized directly from unit information at time of transfer or closing.

Temporal Summation of Event Data

To this point in the paper, events accounting has been viewed in terms of

maximum temporal generality with all transaction data being maintained indefinitely. During design of an actual system, quite obviously, consideration of both decision usefulness and storage costs would temper these requirements somewhat and identify places where temporal summation of event data might be appropriate. An important point to remember about the REA accounting model in this respect is that modifications to the ideal (complete events system) are made knowingly and only after an exhaustive cost-benefit analysis of *all* (not just accounting) data usage has indicated that the adjustments make economic sense. Imperfections are the result of deliberate choice. Such a process is in sharp contrast with traditional accounting where transaction summations are routinely directed by reporting cycles. In REA terms, closing summations are usually derived data or viewed data, and they should be treated as such.

The methods for effecting temporal summations could include any of the procedure types shown in Figure 8. Additionally, it could be decided during the physical database design phase to use a hierarchy of storage devices [Lum et al., p. 91] which would mean that less recent event data could be moved from the prime database to slower and less expensive types of storage media for less frequent use.[5]

Event Partitioning and Combination

An economic event can be defined as a single occurrence in time or as an arbitrary division of a continuous process. It matters not which definition is chosen so long as all users agree upon its specification. Certain accounting conventions make this agreement process difficult, because they take event sets and either combine them or partition them more finely. Some of these conventions are

discussed in the paragraphs below along with methods for accommodating them within the REA model.

Event partitioning in an accounting sense equates with adjusting entries and involves situations where accountants and non-accountants can both agree and disagree on definitions. A case where they probably would agree would be depreciation, a non-exchange event where the definition is admittedly arbitrary (although there are methods for making it less so [Hendriksen, 1977, pp. 419–21]). Cases where they probably would disagree would involve accruals such as wage expense or interest revenue. These accruals are approximate partitions of exchange transactions, and non-accountants are more likely to favor an event definition corresponding to the exchange reality. A procedural solution to this second case would entail specifying the non-accounting event as a base object and maintaining the accrual information as derived data for the category of events affected. The procedures for periodic reporting could then be patterned after those used in conventional adjusting and reversing entries.

Event combinations involve situations where users view event sets that are conceptually different (according to the generalized framework) as congruent. A common accounting example of combination—expensing of *immediate services* [Bedford, 1965, p. 77]—is illustrated in Figure 9(a). Whenever services are acquired and consumed during the same accounting period, the current practice is to treat the two events as one and not to materialize a conclusion about the resource involved. For the advertising example given, this convention would mean that the sets drawn with broken

[5] Methods for having such migration managed automatically (thus making it at least partially transparent to database users) were surveyed recently by Smith [1981].

FIGURE 9

EVENT COMBINATIONS

(a) Expensing of immediate services


(b) Transfer of materials
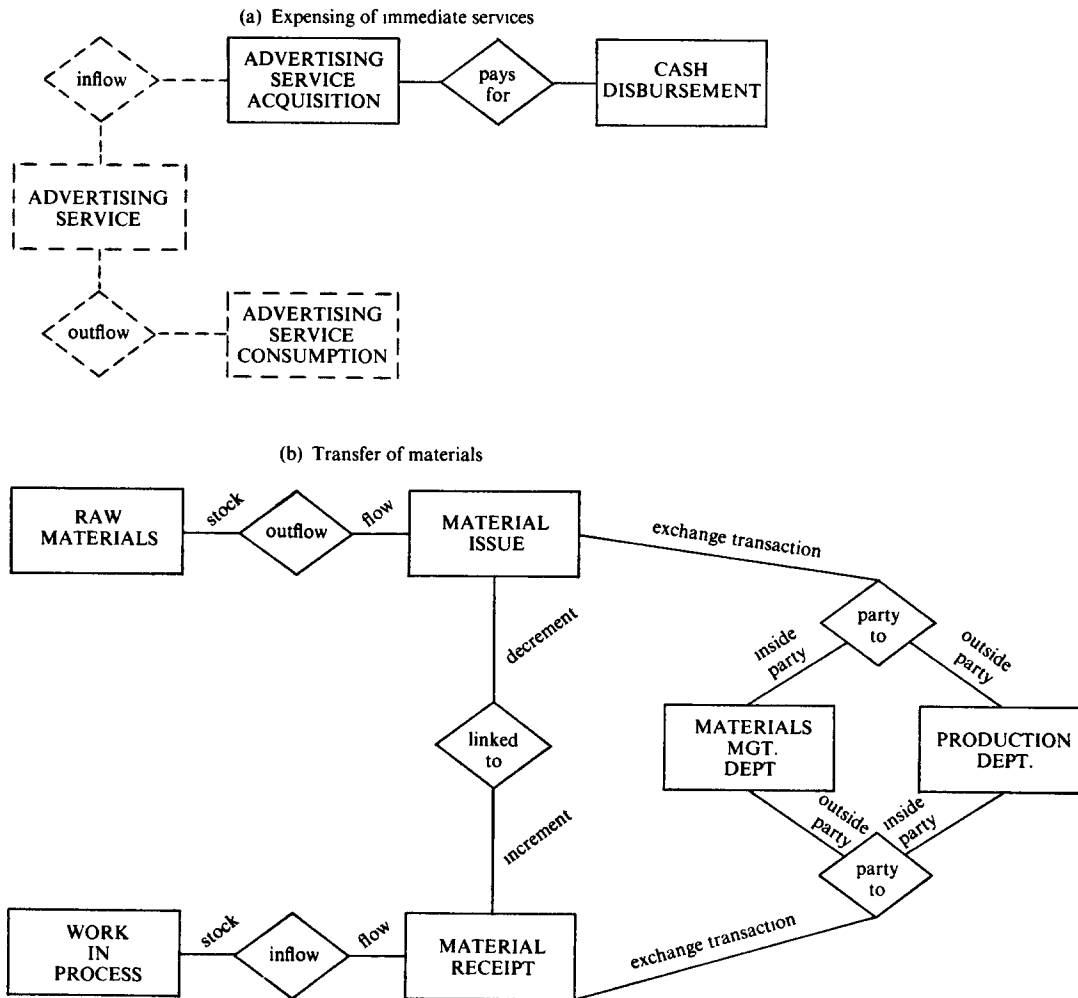
lines would not be needed. A second combination example—transfer of materials from stores to production—is shown in Figure 9(b) (roles have been drawn for this example to facilitate understanding). Current practice in this case would also specify combination because when the events "material issue" and "material receipt" are viewed from an enterprise-wide perspective, they become simple mirror images of each other.

Thus, there exists no reason to view and maintain them separately. As a general rule, combinations in base objects suggested by accounting convention can be implemented if they do not affect the information perspective of other database users.

Macro-level Duality

Duality relationships in the REA model associate each increment in the

FIGURE 10

MATCHING AS MACRO-LEVEL DUALITY
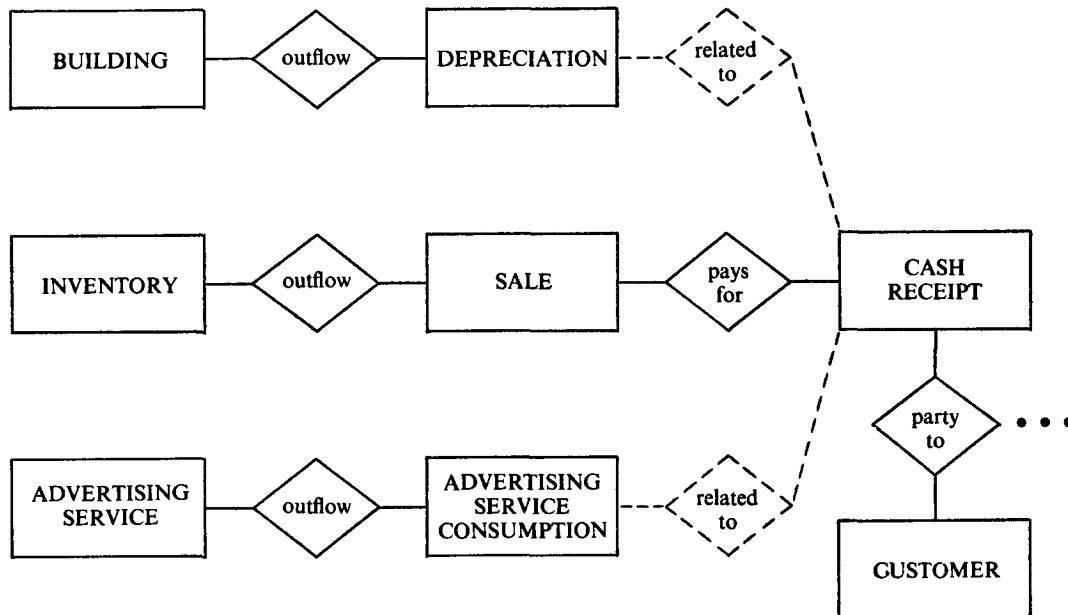
resource set with a decrement and vice versa. As Ijiri notes [1975, p. 84], the mind-set engendered by duality significantly affects the manner in which accountants perceive a system of economic events, because it forces them always to look at resource changes in relation to other changes rather than as isolated occurrences. There are times, however, when these duality requirements may be discarded, at least at the level of individual entity and relationship sets. Two such situations are discussed below: (1) matched expenses, and (2) gains and losses.

"Matching is the process of reporting expenses on the basis of a cause-and-effect relationsip with reported revenues" [Hendriksen, 1977, p. 198]. Two examples that illustrate this convention are shown in Figure 10 with broken lines. The use of matching (the connecting of

"advertising service consumption" and "depreciation" with "cash receipt") and the use of base object sets (the connecting of "sale" with "cash receipt") are simply two different ways of implementing duality links. The important procedural difference between these two methods is that matching occurs only at summarized (macro) levels during financial statement preparation. The reason for this practice is that the relationship between most expense events and the resource increments that they ultimately generate is so tenuous in nature that specific association of the two is neither possible nor desirable.

Gains and losses are resource changes not associated with the normal earning activities of an enterprise. The exact nature of gains and losses is difficult to define and relies to a great extent upon interpretation of existing accounting

practice. For example, some gains and losses arise from exchange transactions and therefore involve both increments and decrements, but there are also many others that involve only nonreciprocal transfers [FASB, 1979, pp. 31–2]. For purposes of schema analysis, it will suffice to note that, for these particular types of accounting phenomena, there are many occasions when increments and decrements occur quite legitimately in isolation.

Equity Transactions

In an REA accounting model, owners' equity represents the imbalance in a duality relationship linking resource investments by owners (increments) with a continuing stream of resource distributions back to those owners (decrements). The exact nature of this imbalance is impossible to identify in going concerns, because the distributions are never fully specified. Therefore, some considerations involved in adapting the REA framework to current conventions for equity accounting are discussed below.

The FASB [1979] defines owners' equity as the residual interest of the owners in the assets of an enterprise. This interest derives from an excess of assets over liabilities and "is the cumulative result of investments by owners, profit, and distributions to owners" [FASB, 1979, p. 65].

Because equity is a residual, its dollar amount at a certain time can be identified procedurally in an REA database by materializing dollar summations of resources and claims. This procedural solution, however, does not provide the transaction information needed to specify the *composition* of owners' equity. To accomplish that task, it is necessary to model owners' investments and distributions explicitly with entity sets such as "equity investment" and "dividend dis-

tribution." These types of capital transactions are similar to those events representing inflows and outflows to claims. That is, if it is necessary to model or derive equity as a stock information object, then it becomes possible to model its flows with events even though the events affect no identifiable resource. The capital transactions could then be associated with increment/decrement events to account for the flows involved in the transfer of resources with owners. For example, "equity investment" could be linked to "cash receipt" for a sale of stock and "dividend distribution" to "cash disbursement" for a dividend payment.

Existing Practice Summary

The paragraphs above have described how the generalized framework can be used to analyze information requirements and how both declarative and procedural elements of data modeling can be tailored to accommodate accounting practices such as accruals and matching. There are other ways in which accounting rules will affect the modeling effort, but as the FASB [1979] points out, many of these practices tend to vary with both conditions of uncertainty and circumstances surrounding a particular enterprise. The accountant will have to decide during design analysis just how the basic entities and relationships are to be interpreted in light of current conventions and pronouncements.

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

The derivation and use of the REA accounting framework has now been described completely. It was argued that application of this entity-based model of accounting phenomena would allow design of a database that could meet the needs of both accounting and non-

accounting users via shared access to an unbiased enterprise schema. The discussion here focused on the specification of that enterprise schema (that is, on requirements analysis, view modeling and view integration); however, readers should note that complete design of a database includes two additional phases: implementation design and physical design [Lum et al., 1979]. Those phases were not discussed because of their heavy dependence upon the particular piece of software being used.

The paper will conclude with a brief discussion of possible research directions both for conceptual modeling of accounting systems in general and for REA modeling in particular.

First, more accounting research should be done that deals with the complex abstraction and modeling issues [Brodie and Zilles, 1981] inherent in the use of modern computer systems. Accountants will have to use these systems, and insights into many of the problems they will face can be found in the computer science literature. These statements apply not just to database theory (where most of the work in accounting has been done thus far) but also to areas such as artificial intelligence, office automation, and decision support systems.

Second, more accounting data models need to be formulated. Additional work is needed to extend REA-type models to areas such as budgeting and commitment accounting [Ijiri, 1975], social accounting, and nonbusiness accounting. Concepts and principles introduced by any of the newer second generation data models could also be used to build on the REA foundation.

Third, many implementation issues need to be explored further. The use of an REA model in the design of relatively complex accounting object systems would lend considerable insight into issues such as: (1) the proper balance between the declarative and procedural features of an accounting database, (2) the amount of redundancy to use for control purposes, and (3) the appropriate way to satisfy the traditional general ledger needs of both financial and managerial accountants. McCarthy and Gal [1981] have explored some of these topics, but the enterprise they modeled was very simple, and more extensive research is needed. Additionally, work needs to be done is assessing how well a shared usage schema fares in integrating accounting systems with other complex information systems that deal with similar economic phenomena but which have traditionally been built and maintained separately (such as systems for logistics/distribution management and material requirements planning).

Finally, the REA accounting model should be used for research in areas of accounting other than those described in the paper. Because of both the framework's generalized nature and the well-formulated results produced by semantic modeling, it might be possible with an REA schema to design comprehensive yet simple solutions to problems such as internal control specification and audit evidence gathering [Weber, 1977]. It should also be possible to perform user validation studies, similar to those published by Benbasat and Dexter [1979], that would compare the REA model against both traditional systems and other database systems.

REFERENCES

Abrial, J. R. (1974), "Data Semantics," in J. W. Klimbie and K. L. Koffeman, eds., *Data Base Management* (North Holland Publishing Company, 1974), pp. 1–60.

American Accounting Association (1969), "Report of Committee on Managerial Decision Models," THE ACCOUNTING REVIEW (Supplement 1969), pp. 43–76.

———— (1971), "Report of the Committee on Non-Financial Measures of Effectiveness," THE ACCOUNTING REVIEW (Supplement 1971), pp. 164–211.

Bachman, C. W., and M. Daya (1977), "The Role Concept in Data Models," *Proceedings of the Third International Conference on Very Large Data Bases* (ACM, 1977), pp. 464–76.

Bedford, N. M. (1965), *Income Determination Theory: An Accounting Framework* (Addison-Wesley, 1965).

Benbasat, I., and A. S. Dexter (1979), "Value and Events Approaches to Accounting: An Experimental Evaluation," THE ACCOUNTING REVIEW (October 1979), pp. 735–49.

Brodie, M. L. and S. N. Zilles, eds. (1981), "Proceedings of the Workshop on Data Abstraction, Databases and Conceptual Modelling," *SIGMOD Record* (February 1981).

Bubenko, J. (1977), "The Temporal Dimension in Information Modeling," in G. M. Nijssen, ed., *Architecture and Models in Data Base Management Systems* (North Holland Publishing Company, 1977), pp. 93–118.

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

Chamberlin, D. D., M. M. Astrahan, K. P. Eswaran, P. P. Griffiths, R. A. Lorie, J. W. Mehl, P. Reisner, and B. W. Wade (1976), "SEQUEL 2: A Unified Approach to Data Definition, Manipulation and Control," *IBM Journal of Research and Development* (November 1976), pp. 560–75.

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

———— (1978), "Applications of the Entity-Relationship Model," *Proceedings of the NYU Symposium on Database Design* (NYU, 1978), pp. 25–33.

Codd, E. F. (1979), "Extending the Database Relational Model to Capture More Meaning," *ACM Transactions on Database Systems* (December 1979), pp. 397–434.

Colantoni, C. S., R. P. Manes, and A. B. Whinston (1971), "A Unified Approach to the Theory of Accounting and Information Systems," THE ACCOUNTING REVIEW (January 1971), pp. 90–102.

Cooper, R. B., and E. B. Swanson (1979), "Management Information Requirements Assessment: The State of the Art," *Data Base* (Fall 1979), pp. 5–16.

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

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

Eswaran, K. P. (1976), "Aspects of a Trigger Subsystem in an Integrated Database System," *Proceedings of Second International Conference on Software Engineering* (IEEE, 1976), pp. 243–50.

Everest, G. C., and R. Weber (1977), "A Relational Approach to Accounting Models," THE ACCOUNTING REVIEW (April 1977), pp. 340–59.

Financial Accounting Standards Board (1979), *Elements of Financial Statements of Business Enterprises*, Exposure Draft (FASB, 1979).

Flavin, M. (1981), *Fundamental Concepts of Information Modeling* (Yourdon Press, 1981).

Haseman, W. D., and A. B. Whinston (1976), "Design of a Multi-dimensional Accounting System," THE ACCOUNTING REVIEW (January 1976), pp. 65–79.

———— (1977), *Introduction to Data Management* (Richard D. Irwin, 1977).

Hammer, M. and D. McLeod (1981), "Database Description with SDM: A Semantic Database Model," *ACM Transactions on Database Systems* (September 1981), pp. 351–86.

Hawryszkiewycz, I. T. (1980), "Data Analysis—What are the Necessary Concepts," *The Australian Computer Journal* (February 1980), pp. 2–14.

Hendriksen, E. S. (1977), *Accounting Theory* (Richard D. Irwin, 1977).

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

Kahn, B. (1978), "A Structured Logical Database Design Methodology," *Proceedings of the NYU Symposium on Database Design* (NYU, 1978), pp. 15–24.

Kent, W. (1978), *Data and Reality* (North Holland Publishing Company, 1978).

Lieberman, A. Z., and A. B. Whinston (1975), "A Structuring of an Events-Accounting Information System," THE ACCOUNTING REVIEW (April 1975), pp. 246–58.

Lum, V., S. Ghosh, M. Schkolnick, D. Jefferson, S. Su, J. Fry, T. Teorey, and B. Yao (1979), "1978 New Orleans Data Base Design Workshop Report," Research Report RJ2554 (IBM Research Laboratories, San Jose, CA, July 1979). (Abridged version also published in *Proceedings of the Fifth International Conference on Very Large Data Bases* (IEEE, 1979), pp. 328–39).

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

McCarthy, W. E. (1979), "An Entity-Relationship View of Accounting Models," THE ACCOUNTING REVIEW (October 1979), pp. 667–86.

———— (1980), "Construction and Use of Integrated Accounting Systems with Entity-Relationship Modeling," in P. Chen, ed., *Entity-Relationship Approach to Systems Analysis and Design* (North Holland Publishing Company, 1980), pp. 625–37.

————, and G. Gal (1981), "Declarative and Procedural Features of a CODASYL Accounting System," in P. Chen, ed., *Entity-Relationship Approach to Information Modeling and Analysis* (ER Institute, 1981), pp. 197–213.

Munro, M. C., and B. R. Wheeler (1980), "Planning, Critical Success Factors, and Management's Information Requirements," *Management Information Systems Quarterly* (December 1980), pp. 27–38.

Mylopoulos, J., P. A. Bernstein, and H. K. T. Wong (1980), "A Language Facility for Designing Database-Intensive Applications," *ACM Transactions on Database Systems* (June 1980), pp. 185–207.

Sakai, H. (1981), "A Method for Defining Information Structures and Transactions in Conceptual Schema Design," *Proceedings of the Seventh International Conference on Very Large Data Bases* (IEEE, 1981), pp. 225–34.

Schiffner, G., and P. Scheuermann (1979), "Multiple Views and Abstractions with an Extended-Entity-Relationship Model," *Computer Languages* (Volume 4, 1979), pp. 139–54.

Smith, A. J. (1981), "Long Term File Migration: Development and Evaluation of Algorithms," *Communications of the ACM* (August 1981), pp. 521–32.

Smith, J. M. (1978), "A Normal Form for Abstract Syntax," *Proceedings of the Fourth International Conference on Very Large Data Bases* (IEEE, 1978), pp. 156–62.

————, and D. C. P. Smith (1977a), "Database Abstractions: Aggregation," *Communications of the ACM* (June 1977), pp. 405–13.

———— (1977b), "Database Abstractions: Aggregation and Generalization," *ACM Transactions on Database Systems* (June 1977), pp. 105–33.

———— (1978), "Principles of Database Design," *Proceedings of the NYU Symposium on Database Design* (NYU, 1978), pp. 35–49.

Sorter, G. H. (1969), "An 'Events' Approach to Basic Accounting Theory," THE ACCOUNTING REVIEW (January 1969), pp. 12–19.

Taggart, W. M., and M. O. Tharp (1977), "A Survey of Information Requirements Analysis Techniques," *Computing Surveys* (December 1977), pp. 273–90.

Teorey, T. J., and J. P. Fry (1980), "The Logical Record Access Approach to Database Design," *Computing Surveys* (June 1980), pp. 179–211.

Tsichritzis, D. C., and A. Klug, eds., (1978), "The ANSI/X3/SPARC DBMS Framework: Report of the Study Group on Database Management Systems," *Information Systems* (Volume 3, 1978), pp. 173–91.

Weber, R. (1977), "Implications of Database Management Systems for Auditing Research," in B. E. Cushing and J. L. Krogstad, eds., *Frontiers of Auditing Research* (The University of Texas Bureau of Business Research, 1977), pp. 207–43.

Wong, H. K. T., and J. Mylopoulos (1977), "Two Views of Data Semantics: A Survey of Data Models in Artificial Intelligence and Database Management," *INFOR* (October 1977), pp. 344–83.

Yu, S. C. (1976), *The Structure of Accounting Theory* (The University Presses of Florida, 1976).