REA
An Evolutionary Perspective

James A. Hall, (E-mail: jah0@lehigh.edu), Lehigh University

Abstract

Controversial features of REA divide AIS educators on the virtues of teaching REA versus traditional AIS models. Some educators on both sides of the issue consider REA and traditional AIS to be mutually exclusive propositions. This article demonstrates that REA is an evolutionary rather than revolutionary concept that is rooted in traditional AIS models. The article examines the evolutionary relationship between REA and the manual process, the flat-file, and the data base models. The article shows that REA and traditional AIS can and should share the same podium. In fact, a sound appreciation of REA principles follows from insight gained by studying and understanding traditional AIS models. The appendix to the article presents a strategy for teaching REA within the structure of a traditional AIS course.

Introduction

Controversial features of REA (Resources, Events, and Agents), such as the abandonment of double-entry accounting, the elimination of journals, and the eradication of the chart-of-accounts have the potential to polarize the AIS community. Camps are forming on both sides of an REA line drawn in the sand. Advocates for REA portray it as the new order that sounds the death knell for traditional AIS education. On the other side, critics suggest that REA's defining features are more semantic than substantive. They argue that REA is simply a product of a mind-set that values change for the sake of change. Extremists on both sides view REA and traditional systems as mutually exclusive propositions that should not share the same podium. Many others in the AIS community, seeking defensible middle ground between these extreme positions, are justifiably confused.

Although sometimes misunderstood, the virtues of REA are real and its role in the future of AIS is likely to grow rather than diminish. REA concepts should be embraced and incorporated into traditional AIS courses. This does not mean abandoning all that transpired in our field before REA nor does it not mean re-engineering AIS education. REA represents evolutionary not a revolutionary concepts. The changes needed in our thinking and in our teaching are incremental not fundamental.

The purpose of this article is to demonstrate that REA and traditional systems evolved from common roots. Placing the REA model in evolutionary perspective along with the manual process, the flat file, and the data base models, provides a foundation for building a better appreciation for the REA philosophy. In addition, such perspective yields an effective framework for educating AIS students. The appendix to this article contains a strategy, based on this perspective, for teaching REA within the structure of a

Readers with comments or questions are encouraged to contact the authors via e-mail.

traditional AIS course.

The Manual Process Model

REA is often thought of as being an information technology solution. Indeed, without advanced technology REA could not be implemented. The REA concept, however, has been around much longer than the technology that supports it. In fact, REA is a conceptual model for describing the physical phenomena (*Resources*, *Events*, and *Agents*) that characterize business processes including manual procedures.

Resources. Economic resources are the assets of the organization. They are defined as objects that are both scarce and under the control of the enterprise. These include materials, financial capital, and information.

Events. Economic events are phenomena that affect changes in resources. They can result from activities such as production, exchange, consumption, and distribution. Events driven information should be captured in a highly detailed form to provide a rich data base.

Agents. Agents are individuals and groups that participate in an economic event. They are parties both inside and outside the organization with discretionary power to use or dispose of economic resources. Examples of agents include sales clerks, production workers, shipping clerks, customers, and vendors.

The Manual Process Model in REA Evolution

An objective of REA is the redesign of traditional business processes to achieve improvements in performance. This includes an events-driven accounting system that supports the information needs of all users in the organization rather than the needs of accountants exclusively. Such systems are not designed in a vacuum from scratch. They typically evolve from more traditional systems. As a practical matter, therefore, understanding the business procedures, operations, information flows, and internal controls that constitute the manual process model is essential to a successfully re-engineered REA system.

The Flat-File Model

The flat-file model describes an environment in which individual data files are not related to other files. End users in this environment *own* their data files rather than *share* them with other users. Data processing is thus performed by stand-alone applications rather than integrated systems.

When multiple users need the same data for different purposes, each must be provided with separate data sets structured to their specific needs. Figure 1 illustrates how customer sales data might be presented to three different users in a durable goods retailing organization. The Accounting function needs customer sales data organized by account number and structured to show outstanding balances. This is used for customer billing, account receivable maintenance, and financial statement preparation. Marketing needs customer sales history data organized by demographic keys. They may use this for targeting new product promotions and for selling product upgrades. The Product Service function needs customer sales data organized by products and structured to show scheduled service dates. Such information is used for making after-sales contacts with customers to schedule preventive maintenance and to solicit sales of service agreements.

The data redundancy demonstrated in this example contributes to three general problems of significance in the flat-file environment: data storage, data updating, and currency of information.

Data Storage

An efficient information system captures and stores data only once and makes this single source available to all users who need it. In the flat-file environment, this is not possible. To

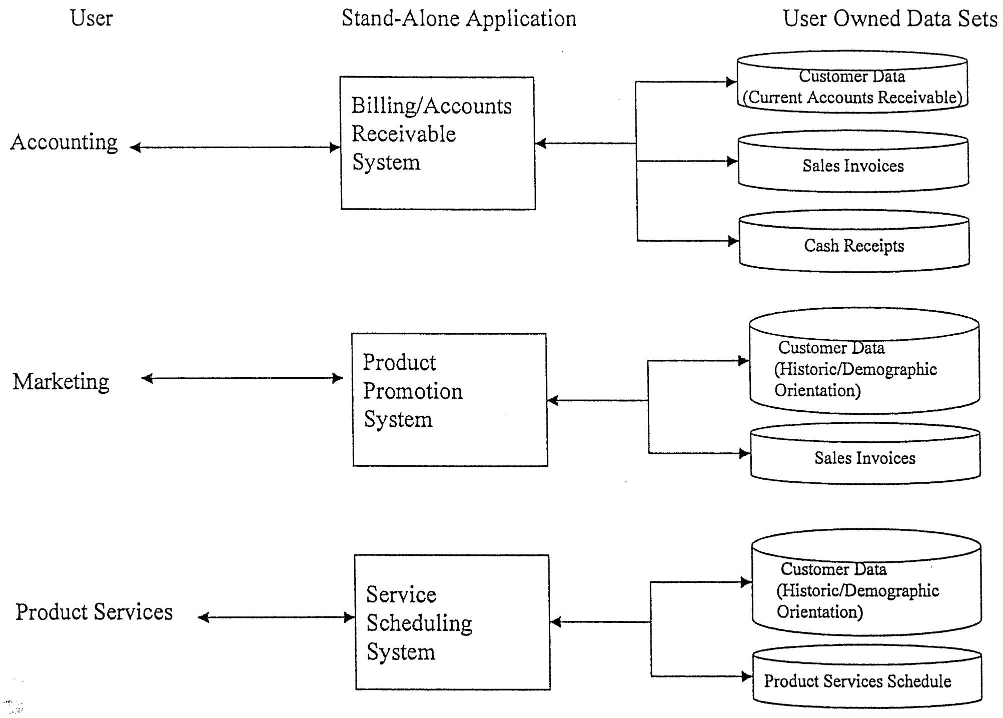

User
Stand-Alone Application
User Owned Data Sets
Accounting
Billing/Accounts Receivable System
Customer Data (Current Accounts Receivable)
Sales Invoices
Cash Receipts
Marketing
Product Promotion System
Customer Data (Historic/Demographic Orientation)
Sales Invoices
Product Services
Service Scheduling System
Customer Data (Historic/Demographic Orientation)
Product Services Schedule

Figure 1, Flat-File Model

meet the private data needs of users, organizations must incur the costs of both multiple collection and multiple storage procedures. Some commonly used data may be duplicated dozens, hundreds, or even thousands of times.

Data Updating

Organization's have a great deal of data stored on master files and reference files that require periodic updating to reflect changes. For example, a change to a customer's name or address must be reflected in the appropriate master files. When users keep separate files, any such change must be made separately for each user. This adds significantly to the task burden and the cost of data management.

Currency Of Information

In contrast to the problem of performing multiple updates, is the problem of failing to update all the user files effected by a change in status. If update information is not properly disseminated, the change will not be reflected in some users data resulting in decisions based on outdated information.

Flat-Files and Integration

The flat-file approach is a single view model. Data files are structured, formatted, and arranged to suite the specific needs of the *owner* or primary user. Such artificial structuring, however, may destroy data attributes that are essential to other users thus preventing successful integration of systems across the organization.

For example, since the accounting function is the primary user of accounting data, these data are often captured, formatted, and stored to accommodate financial reporting and GAAP.

This data structure may, however, be useless to other (non accounting) users of accounting data such as the marketing, finance, production, and engineering functions of an organization. These users are faced with three options: 1) do not use accounting data to support decisions; 2) manipulate the existing data structure to suite their unique needs; or 3) obtain additional private views of the data and incur the costs and operational problems associated with data redundancy.

Flat-Files in REA Evolution

The single-view flat-file approach is the antithesis of the multiple view REA model. REA describes an environment in which users share data rather than own private views of it. This requires that data are stored in a pristine format that is not dominated by a single user's view and, which can support the needs of multiple users. Although the flat-file and REA models represent endpoints on an integration continuum, the evolutionary linkage between them is clear. The failings of the traditional flat-file model are the very issues that engendered the REA concept. Studying and understanding these issues stimulates a sound appreciation of REA as a potential solution for the many legacy systems that have yet to be re-engineered and integrated. The technology model that supports the REA solution is discussed next.

The Data Base Model

Figure 2 presents a simple overview of the data base approach. The most striking change from the flat-file model is the pooling of data into a common data base that is shared by all organizational users. Users have access to the full domain of data available to the firm. As their information needs change and mature they can be satisfied without obtaining additional private data sets. Users are constrained only by the limitations of the data available to the entity and the legitimacy of their need to access it. Three aspects of the data base model that contribute directly to an understanding of REA are the *data base philosophy*, the concept of a *user view*, and *data normalization*. Each of these is discussed below.

The Data Base Philosophy

The data base model is not simply a data management technology. It is an integrative data management philosophy with objectives consistent with REA objectives. Through data sharing the traditional problems associated with the flat-file approach *may* be overcome.

No *data redundancy*. Each data element is stored only once, thereby eliminating data redundancy and reducing data collection and storage costs. For example, customer data exists only once, but is shared by accounting, marketing, and product services users.

Single update. Because each data element exists in only one place, it requires only a single update procedure. This reduces the time and cost of keeping the data base current.

Current values. A single change to a data base attribute is automatically made available to all users of the attribute. For example, a customer address change entered by the billing clerk is immediately reflected in the marketing and product services views.

Improved Integration. The data base model permits data to be linked to other related data. For example, sales, marketing, production, inventory, and supplier data could all be logically linked to achieve integration among users in different functional areas.

The use of data base technology, however, does not in itself guarantee an integrated system. If not employed within the context of the REA framework, this technology may be used simply to replace flat-files, thus perpetuating traditional thinking and traditional problems.

The User View

A user view is the set of data that a par-

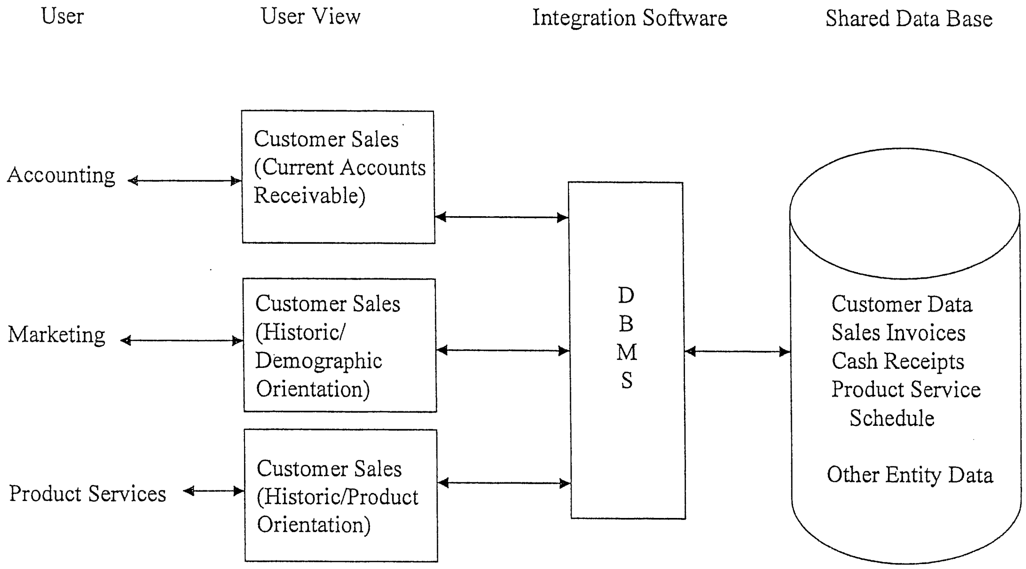


Figure 2, Data Base Model

ticular user needs to achieve his or her assigned tasks. User views across an entity are diverse. To illustrate: a general ledger clerk's view consists of the organization's chart-of-accounts; a sales manager's view might include detailed customer sales data organized by product, region, and sales person; and a production managers view may include finished goods inventory on hand, available manufacturing capacity, and vendor lead times.

A problem arises in meeting diverse user needs when the collection, summarization, storage, and reporting of transaction and resource data are dominated by a single view that is inappropriate for entity-wide purposes. Traditionally, the accountant's view has dictated the set of accounting data used by organizations. However, modern managers need both financial and non-financial information in formats and at levels of aggregation that are different than traditional accounting systems can accommodate.

Only when the physical data base is constructed in sufficient detail can data base management technology sustain diverse user views. The desire to support the needs of all users, therefore, must become the data base strategy for any firm pursuing REA.

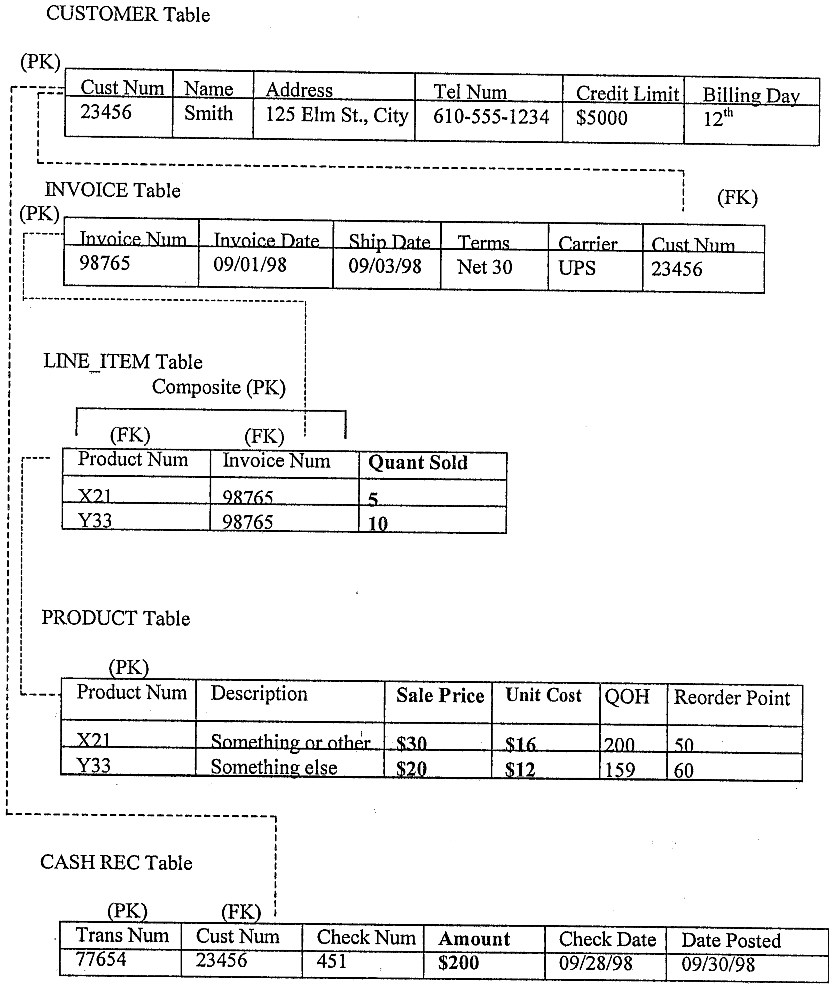
Data Normalization

A brief review of data normalization demonstrates why double entry accounting, journal entries, and ledgers may no longer exist in the traditional sense. The following revenue cycle transactions for a hypothetical retailer illustrate this point:

Sept 1: Sold 5 units of product X21 @ $30 per unit and 10 units of product Y33 @ $20 per unit to customer Smith. The unit cost of the inventory is $16 and $12 respectively.

Sept 30: Received $200 cash from customer Smith on account, check number 451.

Recording these events under the data base approach would require the normalized base tables shown in Figure 3. The primary and foreign keys



CUSTOMER Table

Cust Num	Name	Address	Tel Num	Credit Limit	Billing Day
23456	Smith	125 Elm St., City	610-555-1234	$5000	12th

INVOICE Table

Invoice Num	Invoice Date	Ship Date	Terms	Carrier	Cust Num
98765	09/01/98	09/03/98	Net 30	UPS	23456

LINE_ITEM Table

Product Num	Invoice Num	**Quant Sold**
X21	98765	5
Y33	98765	10

PRODUCT Table

Product Num	Description	**Sale Price**	**Unit Cost**	QOH	Reorder Point
X21	Something or other	$30	$16	200	50
Y33	Something else	$20	$12	159	60

CASH REC Table

Trans Num	Cust Num	Check Num	**Amount**	Check Date	Date Posted
77654	23456	451	**$200**	09/28/98	09/30/98

Figure 3, Normalized Base Tables With Embedded Keys

linking the tables are illustrated by dotted lines. Normalized data base tables permit the capturing of transaction data in great detail. Each normalized table deals with a separate aspect of the transaction. Data pertaining to the customer, the invoice, the specific items sold and so on, can thus be captured for multiple users and uses. Flat-file systems usually do not accommodate such detailed data gathering. Instead, the transaction would need to be summarized into a traditional accounting view at the loss of potentially important facts.

Notice that in this example, traditional accounting records including journals, ledgers, and charts-of-accounts do not exist as physical tables. Accounting records can, however be represented as virtual tables that are constructed from the above base tables to create user views. For example, the amount of Smith's Account Receivable balance can be derived from {total sales (Quant sold * Sale Price) less cash received (Amount) = 150}. If necessary, journal entries and general ledger amounts can also be derived from these base tables. For example, the Cost-of-Goods-Sold Control balance is (Quant sold *Unit cost) summed for all transactions for the period.

Data Base Model in REA Evolution

The departure from familiar bookkeeping techniques has, for some, been an impediment to embracing the REA philosophy. The point of the example above is to demonstrates that the abandonment of traditional accounting records, often attributed as one of REA's more egregious features, is actually a characteristic of normalized data base design. If an accounting system with a more traditional look is desired, however, physical journal tables, subsidiary ledger tables, and chart-of-accounts tables can be constructed in addition to the detail base tables above. While this action would result in some data redundancy, it would not damage the integrity of the underlying detail data base and would not nullify the objectives of REA.

The REA Model

REA is an accounting framework for modeling an organization's critical *resources*, *events*, and *agents* (REA) and the relationships between them. Once this is done, both accounting and non accounting data about these phenomena can be identified, captured, and stored in a centralized data base. From this repository, user views can be constructed that meet the needs of all users in the organization. The availability of multiple views allows flexible use of transaction data and permits the development of accounting information systems that promote, rather than inhibit, integration.

McCarthy (1982) originally proposed REA as a theoretical model for accounting. Advances in data base technology have focused renewed attention on REA as a practical alternative to the traditional accounting framework. The REA model requires that accounting phenomena be characterized in a manner consistent with the development of multiple user views. Business data must not be pre formatted or artificially constrained and must reflect all relevant aspects of the underlying economic events. As such, REA data modeling does not include traditional accounting elements such as journals, ledgers, charts-of-accounts, and double entry accounting. According to the model, REA organization's produce their financial statements directly from the transaction detail data base.

Eliminating the Chart-of-Accounts

To many, both among the REA rank-and-file and its skeptics, the preparation of financial statements directly from transaction details, thus eliminating the chart-of-accounts, is REA's defining characteristic. This controversial issue has unfortunately overshadowed REA's many virtues and may have inhibited its widespread acceptance.

The elimination of the chart-of-accounts should not be the defining characteristic of REA and is, in some cases, an unnecessary departure from accounting convention. An accounting system that functions without a chart-of-accounts demonstrates only that the financial transactions data base is sufficiently detailed to prepare financial statements. This does not imply that the organization's information system supports multiple user views with financial and non-financial data nor that it promotes improved decisions and efficient operations through integration.

On the other hand, the presence of a chart-of-accounts does not necessarily signify an information system that reflects a narrow traditional perspective. Some REA organization's may benefit by preparing financial statements directly from transaction details while others may not. The decision to retain the chart-of-accounts may be influenced by factors such as line of business, organizational structure, size, and transaction volume. For example, the management of a large organization may determine that preparing financial statements from 25 million transactions is inefficient or problematic. They decide instead to periodically summarize transactions from their detail data base into a traditional chart-of-accounts structure that better facilitates financial reporting. Since the transaction details remain intact and uncorrupted by this act, the objectives of REA are not compromised.

In Summary

Controversial features of REA divide AIS educators on the virtues of teaching REA versus traditional AIS models. Some educators on both sides of the issue consider REA and traditional AIS to be mutually exclusive propositions. This article demonstrates that REA is an evolutionary rather than revolutionary concept that is rooted in traditional AIS models. REA and traditional AIS can and should share the same stage. In fact, an appreciation for REA requires insight that comes from studying and understanding the limitations of traditional AIS models.

Appendix

A Strategy For Teaching REA

In developing an REA teaching strategy, the AIS instructor must consider the incoming student's background and educational experience. Often students at this stage have a narrow perspective of accounting and its role in business. They are comfortable with traditional double entry accounting, journal entries, ledgers, and financial statements. On the other hand, they know little or nothing about business processes and non-traditional uses of accounting information. The AIS course is an opportunity to broaden the students perspective. Premature immersion in REA, however, can be overwhelming rather than enlightening. Moving forward too vigorously can produce irreconcilable differences between REA and traditional accounting causing students to perceive AIS as esoteric material that lies outside mainstream accounting. This appendix presents a teaching strategy based on the evolutionary approach that carries students systematically, logically, and successfully from traditional manual systems to REA.

Teaching the Manual Process Model

There is merit in teaching students the manual process model before teaching computer-based systems. First, the study of the manual systems helps establish an important link between the AIS course and traditional accounting. The AIS course is often the first accounting course in which students see where data originate, how they are collected, and how and where information is used to support day-to-day operations. By emphasizing information flows, key tasks, and the role of traditional accounting records in transaction processing, the students' narrow bookkeeping perspective is transformed to a business processes perspective.

Secondly, students are better able to understand the logic of a business process when it is not shrouded by technology. The information needed to trigger and support events such as selling, warehousing, and shipping is fundamental and independent of the technology that underlies the information system. For example, a shipping notice, informing the billing process that a product has been shipped, serves this purpose whether it is produced and processed manually or electronically. When students understand what tasks need to be performed, they are better equipped to explore different and better ways of performing the task using technology.

Finally, manual procedures facilitate teaching internal control activities including segregation of functions, supervision, independent verification, audit trails, and access controls. Since human nature lies at the heart of many internal control issues, we should not overlook the importance of this aspect of the information system.

Teaching the Flat File Model

There are a number of justifications for teaching the flat-file model. One feature of flat-file data management is that it emulates traditional manual record keeping techniques fairly closely thus easing the students transition from manual to automated transaction processing systems. Students quickly see how transaction files and master files relate to traditional accounting records such as source documents, journals, subsidiary ledgers and general ledger accounts. A key learning objective for students is to understand computer-based transaction processing techniques including batch, real-time, and point-of-sale systems. The use of complex data base structures at this point is unnecessary to this objective and can inhibit the learning process.

The legacy system problem is another reason for teaching the flat-file model. Approximately 20 billion lines of COBOL code are executed daily. Many large organizations still use flat-files for their general ledger and other financial systems. Legacy systems continue to exist because they add value for their users and will not disappear until they cease to add value. Students who may have to work with these systems in practice should be aware of their key features.

Finally, it is sometimes useful to learn the wrong way of doing things. Upon guided examination, the shortcomings of the flat-file approach become glaringly apparent to students. They discover how limitations inherent to flat-file structures make the gathering of detailed transaction data difficult or impossible. To facilitate financial reporting and other accounting purposes, transaction data must be aggregated thus making it unusable to non-accounting users. Since data sharing is often impossible, users collect, process, and store data independently under the flat-file model. By studying the traditional problems of exclusive data ownership, data redundancy, data updating, data inconsistency, and sub optimal decision making, students are sensitized and receptive to the virtues of the data base model.

Teaching the Data Base Model

The data base model provides the student with an effective transition between the single view environment of the flat file model and the multiple view REA model. The traditional problems of data storage, data updating, currency of information, and lack of integration are solved by data base technology. The data base model also offers practical explanations for controversial aspects of REA. By studying data normalization techniques, students understand why traditional accounting artifacts including journals, ledgers, and double entry accounting may no longer serve a formal purpose in the data base environment.

This point can be reinforced by a data normalization assignment. For example, the instructor may ask the class to produce on paper the normalized base tables (attributes only) for a traditional accounting user view such as a sales invoice, a purchase order, or a receiving report. Each of these views has repeating groups data that cannot be captured in a single table such as an account receivable record. The exercise can be taken a step further by having the students create the normalized tables for a user view that incorporates both accounting and non accounting data. Such assignments help students reach beyond the traditional boundaries of accountants and consider the accountant's role as an entity-wide information provider.

Teaching the REA Model

At this point the groundwork has been laid to present the prime objectives of REA. Students understand business processes, the shortcomings of traditional systems, and the virtues of data sharing. The need for systems designed to support multiple user views with financial and non financial data, and the need to capture and store business transactions in pristine condition (i.e. uncorrupted by the unique needs of accountants) are logical extensions of concepts taught earlier.

A data-modeling project that allows students to employ REA principles is an excellent reinforcing tool. The student-generated case technique (Greenstein and Hall 1996) can be used for this purpose. Under this approach, rather than solving an existing case in the traditional way, students create their own cases. The assignment constitutes a significant term project on which the students work in teams of two or three. Students develop their cases in two parts. The first is a written scenario describing a hypothetical organization's business processes and its current traditional information system. The students are required to embed into this scenario significant but plausible efficiency and internal control problems. Designing plausible errors into a case demands a high level of understanding and draws upon the students creative and analytical skills. This aspect of the case can be administered to students early in the AIS course when they are studying manual business processes.

The second part of the case is the case solution. Students first document the current system using data flow diagrams, document flowcharts, and system flowcharts. Next they analyze the efficiency and control problems embedded in the case. Finally they re-engineer the current system to employ advanced technologies and techniques. As part of the redesign, students prepare a data model of their firm's business processes using ER diagrams. They are required to specify both the financial and non-financial data attributes for each entity (resource, event, and agent) identified in the ER diagrams. Finally, using Microsoft's Access (or an equivalent package), they create the queries, and normalized base tables to produce a selected number of user views.

The student-generated-case approach can yield surprisingly high quality cases. Portions of cases completed in one semester can be used as flowcharting and internal control examples and excises in subsequent semesters. Thus each semester's REA project generates a supply of mini cases and assignments for use in following semesters. 🕮

References

1. Greenstein, Marilyn M. and Hall, James A. 1996. Using student-generated cases to teach accounting information systems. *The Journal of Accounting Education* (Winter): 493-514.
2. McCarthy, William E. 1982. The REA accounting model: A generalized framework for accounting systems in a shared data environment. *The Accounting Review* (July): 554 -577.