ISSUES IN ACCOUNTING EDUCATION
Vol. 18, No. 4
November 2003
pp. 427-441

The REA Modeling Approach to Teaching Accounting Information Systems

William E. McCarthy

ABSTRACT: The REA model was first conceptualized in a paper for the 1982 *The Accounting Review* as a framework for building accounting systems in a shared data environment, both within enterprises and between enterprises. The model's core feature was an object pattern consisting of two mirror-image constellations that represented semantically the input and output components of a business process. The REA acronym derives from that pattern's structure, which consisted of economic Resources, economic Events, and economic Agents.

Simultaneous with its research publication, REA began to be used as a framework for teaching accounting information systems (AIS), originally at Michigan State University and then gradually at other colleges and universities. In its extended form, the REA model integrates the teaching of accounting transaction structures, commitment and business policy specification, business process engineering, and enterprise value chain construction. As of 2003, REA modeling is used in a variety of AIS courses and featured in a variety of AIS textbooks, both in the United States and internationally.

Keywords: REA accounting model; semantic modeling of accounting phenomena; accounting information systems; REA enterprise ontology.

ORIGINS OF THE REA APPROACH

We shall define a database as the model of an evolving physical world.

—Abrial (1974, 3)

When I first read this quote in a computer science graduate class at the University of Massachusetts (UMASS) in 1975, I was immediately struck by its applicability to accounting transaction-processing systems. The world of database theory was then in the middle of its golden age. Codd's ideas on relational systems (1970, 1972b) and syntactic normalization (1972a) had proven superior in most conceptual and logical respects to the older classes of implementation platforms such as network and hierarchical systems, and Abrial's (1974) paper had signaled the advent of the semantic database era that would culminate soon thereafter with the publications of Chen's (1976) entity-relationship model and Smith and Smith's (1977) data abstraction ideas.

William E. McCarthy is a Professor at Michigan State University.

I acknowledge the long-term help of my nine doctoral students whose hard work in shaping REA model components, in exploring their use in practice, and in finding new ways to use them for AIS instruction have inspired my own constant improvement efforts: Howard Armitage, Paul Steinbart, Graham Gal, Eric Denna, Steve Rockwell, Guido Geerts, Cheryl Dunn, Julie Smith David, and Greg Gerard. Additional acknowledgments are due to Steve Zeff, Severin Grabski, Anita Hollander, Tracey Sutherland, and Bob Haugen.

It became clear to me at UMASS that the accounting databases of the future would be relational and that they would be designed with embedded semantics and strict typing. Over the course of the next five to six years, I published a number of papers (McCarthy 1978, 1979, 1980, 1982) on relational accounting systems with strong semantics, a research stream that culminated with *The Accounting Review* presentation of the REA model in 1982. REA was conceptualized there as a framework for building accounting systems in a shared data environment, both within enterprises and between enterprises. The model's core feature was an object pattern consisting of two mirror-image constellations that represented semantically the input and output components of a business process. The REA acronym derives from that pattern's structure, which consisted of economic Resources, economic Events, and economic Agents. That semantic framework has now been extended to include abstract specification of future resource commitments along both supply and value chains, and in a sense, the original notions of Abrial (1974) (as captured in the previous quote) have been applied to accounting and extended to the following:

> An accounting database is a model of the reality surrounding an evolving business enterprise, including its past set of accountability transactions, its present set of commitments and claims, and its future set of plans and policies.

REA as it stands in 2003 is a vastly extended framework, and it is actually a candidate model for several e-commerce transaction standards (David et al. 2002; Geerts and McCarthy 2003). Additionally, REA modeling is used in a large number of AIS courses and featured in a variety of AIS textbooks, both in the United States and internationally. Some limited use of REA has also been made in both elementary accounting courses and introductory MIS (Management Information Systems) database classes (David, Maccracken, and Reckers 2003; Trimmer et al. 2002).

This paper reviews the elements and ideas of REA modeling in elementary terms and illustrates how those notions are used in teaching accounting information system courses. The second section of the paper reviews the essential features of relational databases and semantic modeling. Following that, I review with examples the essential components of three separate stages of the REA model's growth as it has been used in teaching: (1) its set of basic features; (2) its upward abstraction to enterprise-wide value chains, and its downward decomposition to workflow and task specification; and (3) its extensions into commitments and types. I follow this elementary enumeration with a section where I explain how I use REA to teach my own courses, both at Michigan State University (MSU) and at the American Accounting Association's (AAA) summer workshops for beginning AIS teachers. I finish with some suggestions for connecting with the growing community of AIS scholars who feature semantic modeling of accounting phenomena as a core component of their teaching.

RELATIONAL DATABASES AND SEMANTIC MODELING

Commercial database systems of the present, including those available for both smaller systems (Microsoft® Access) and larger systems (Oracle client-server editions), overwhelmingly tend to favor relational structures. Codd (1970) first envisioned these more than 30 years ago in one of the most important computer science papers ever written. Relational systems structure data as flat tables as seen in Figure 1, where information concerning a limited set of painters and paintings is displayed.

The Painter table has two column headings and six rows of actual data representing six artists; the Painting table also has two column headings with eight rows representing some of the work done by those artists. The column headings are most frequently called *attributes*. The underlined NAME headings represent *primary keys*; attributes whose values represent instances of real-world objects in the database. There is a data row for each primary key, and all other cells in that row are allowed to contain at most one value (that is, multivalued attributes are not allowed in a single cell under normalization constraints). This prohibition against "repeating groups" of attributes keeps the tables flat, and it differentiates relational systems from the variable length record structures of the past.

FIGURE 1
Two Relational Tables

Painter

PAINTER NAME	BIRTH YEAR
Van Gogh	1853
Morisot	1841
Magritte	1898
Picasso	1881
Hopper	1882
daVinci	1452

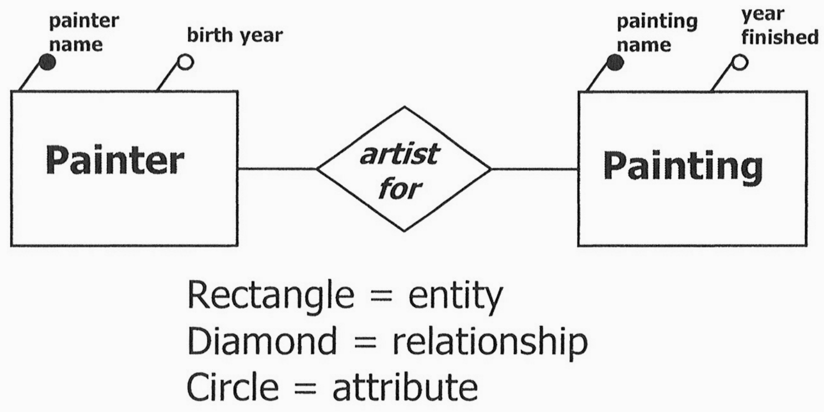
Painting

PAINTING NAME	YEAR FINISHED
Starry Night	1889
The Cradle	1872
On the Balcony	1872
Time Transfixed	1939
Guernica	1937
Nighthawks	1942
The Last Supper	1497
Mona Lisa	1506

The double-sided arrows of Figure 1 informally illustrate which painters and paintings are linked, but they are not part of the relational database mechanics. The connections between tables in an actual implementation are affected by storing different key values in the same row, thus showing that two instances (like "Picasso" and "Guernica") have a relationship with each other. For the database in Figure 1, I could build an association between painters and paintings by either (1) posting the key of Painter into the Painting table (thus giving a third column), or (2) posting the keys of both tables together into a third table (called "artist for") that represents their relationship instances separately. The efficient and effective choice among different posting options is often a topic of AIS classes, but it is beyond my scope here.

The original work on the theory of *normalization* was a bottom-up approach to design, because it structured databases by decomposing large tables into smaller ones by syntactic dependency rules (Everest and Weber 1977). By contrast, the top-down or semantic approach to design—first pioneered fully by Chen (1976)—started with an analysis of the reality being modeled for its various types of objects (like people, things, activities in time, and concepts), and it then proceeded by representing those identified *entities* along with their named links or *relationships* in tabular form (McCarthy 1979). A semantic database designer would always produce an entity-relationship diagram like that shown in Figure 2 before he or she would proceed with the table design options of Figure 1. It is in this arena of top-down modeling where the REA accounting model provides its greatest teaching advantage. That approach is discussed next.

FIGURE 2
Entity-Relationship Notation



COMPONENTS OF THE REA MODEL

The Basic REA Pattern

In a semantic database design (and also in its closely related analysis cousin of object-oriented design), the hardest step is always the first: coming up with a good list of candidate entities (or objects or classes) on which to base the rest of the analysis. The artist example of Figure 2 was relatively simple; modeling business enterprises is much harder. To overcome this difficulty, the *analysis patterns* movement was born (Fowler 1997; Hay 1996; Coad 1995) in the early 1990s. The REA accounting model preceded this work by a decade, but its basic framework of interlocking constellations of Economic Resources, Economic Events, and Economic Agents was actually a complex aggregation of some of those patterns that surfaced in the '90s. The biggest difference was that the REA pattern had much stronger theoretical underpinnings, because it had to pass peer review in the leading journal of its home business discipline (accounting).

Figure 3 illustrates the model in its most basic form (McCarthy 1982, 564) as it exists from the perspective of a business entrepreneur. REA is a pattern for an arm's length collaboration (or alternatively, an inside transformation) between the entrepreneur and a trading partner wherein he or she gives up control of some resource of value (the "give" half of the exchange above the dotted line) in exchange for another resource of perceived greater value (the "take" half of the exchange below the dotted line). The entity types of Figure 3 (the Rs, the Es, and the As) are indeed important, but the structuring effects of the relationships are nearly as paramount (Dunn and McCarthy 1997). *Stock-flow* relationships associate the flows in and out of a resource category while the *duality* links keep the economic rationale for the exchange boldly in the forefront.

Figure 4 illustrates a revenue cycle example of the REA pattern wherein the entrepreneur is a cookie maker who sells his or her wares to a customer[1] who pays later and whose dealings with the entrepreneur are serviced entirely through different classes of the entrepreneur's employees (salesperson and cashier). The progression from Figure 3 to Figure 4 illustrates the type of REA problem I most often have my own students solve on examinations. I try to teach them the pattern of Figure 3; my exams then give them a narrative and attribute list for an example company, and ask that they produce a solution like Figure 4.

[1] In an actual design problem, there would be just one "customer" entity on Figure 4. I have left the second one in to keep Figures 3 and Figure 4 consistent.

FIGURE 3
The REA Pattern

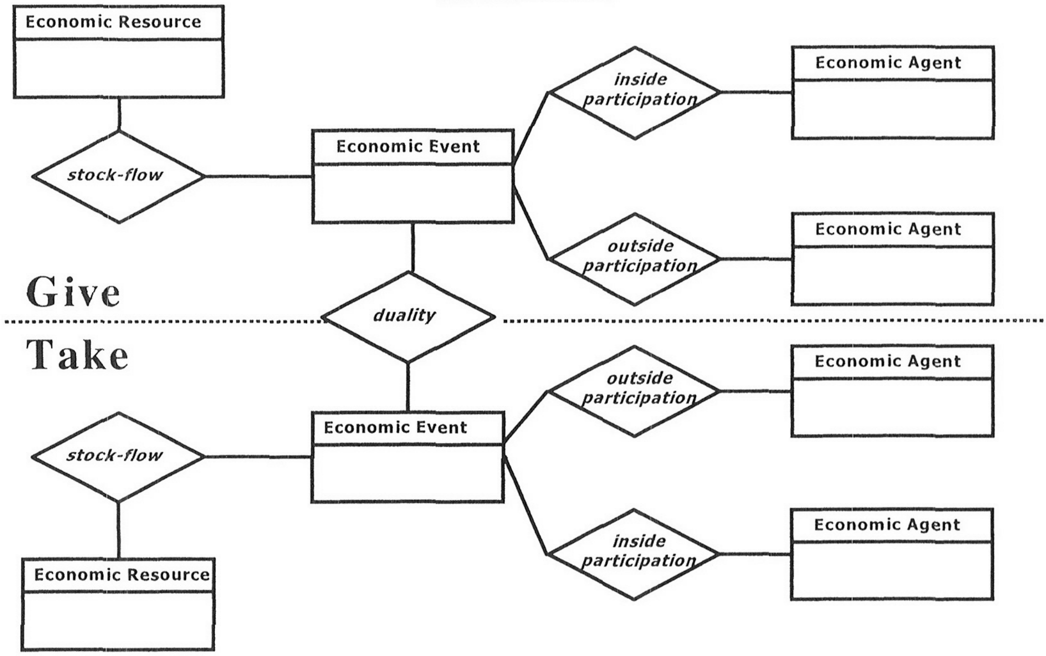


FIGURE 4
REA Revenue Cycle Example

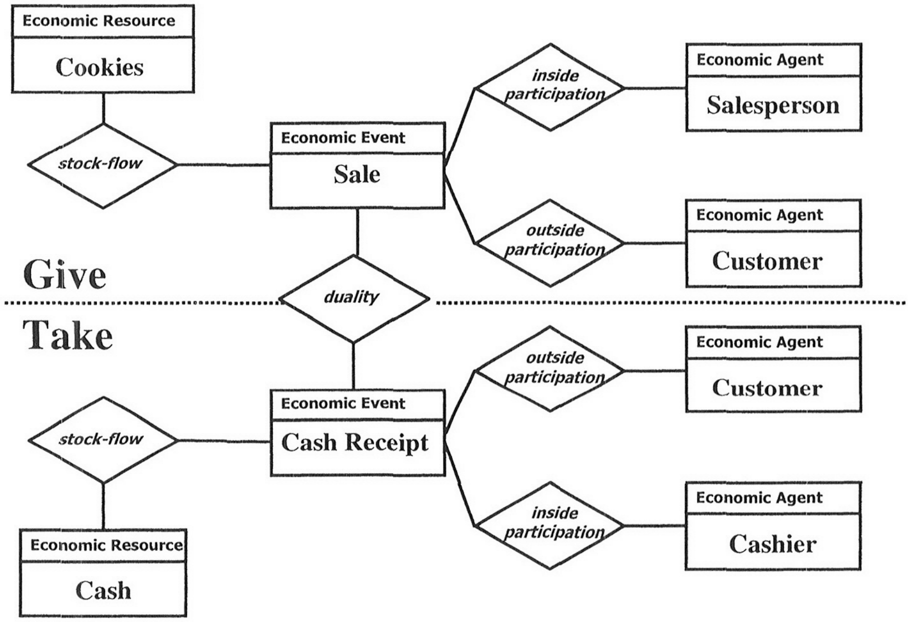


Figure 5 illustrates instances of the revenue cycle model as they would actually be stored in an enterprise-wide relational database. As I promised in the top-down discussion of deriving the tables of Figure 1 from the diagrams of Figure 2, I have fashioned the four tables of Figure 5 directly from two of the entities and two of the relationships of Figure 4 (space considerations preclude mapping the rest).

At this point, accounting readers who are not technology teachers familiar with the intricacies of relational databases can actually check their understanding of the partial data in Figure 5 to see if it makes business sense. For instance, here are some simple accounting and marketing questions:

- Why is the dollar amount of the sale on invoice I-1 set at $14.75?
- How did customer C-888 pay for all of the peanut butter cookies she bought on the 2nd of July?
- Does customer C-999 seem to be showing any patterns in his cookie purchases?
- What are the individual accounts receivable?

I have found these types of questions to be especially effective in demonstrating to students in sophomore-level principles of accounting classes how they can derive accounting numbers from REA tables. When there is time and technology available, I also find it worthwhile to implement such a small set of tables in a database system like Microsoft® Access and let students derive their answers by using a query language.

Readers should also understand that databases with all enterprise cycles complete and fashioned after the methods used to derive Figure 5 could be used to materialize all of the financial accounting numbers needed for external reporting purposes. The methods used to accomplish this are described in a number of theoretical and implementation-oriented papers, all of which illustrate techniques for deriving a *virtual general ledger* from an REA database (McCarthy 1982; Gal and McCarthy 1983, 1986; David et al. 2001). Such a task is always my final computer project in my junior-level AIS course. At MSU, we have developed a special company database for these purposes called *Ventura Vehicles* (David and McCarthy 2003) that any AIS instructor may request for use.

FIGURE 5
Partial Database for Revenue Cycle

COOKIES

Product#	Description	Price	QOH
P-1	Chocolate Chip	1.05	200
P-2	Chocolate	.95	205
P-3	Peanut Butter	1.00	97
P-4	Pecan	1.10	257

COOKIES-stockflow-SALE

Product#	Invoice#	Quantity
P-2	I-1	5
P-3	I-1	10
P-3	I-2	20
P-4	I-3	9
P-1	I-4	4
P-3	I-4	5

SALE

Invoice#	Dollar Amount	Date	Salesperson Employee#	Customer #
I-1	14.75	1JUL	E-1234	C-987
I-2	20.00	2JUL	E-1235	C-888
I-3	9.90	3JUL	E-1236	C-999
I-4	7.20	5JUL	E-1237	C-999

SALE-duality-CASH_RECEIPT

Invoice #	Receipt Timestamp	Amount Applied
I-1	2JUL0830	14.75
I-2	3JUL0800	2.00
I-2	5JUL0800	18.00
I-3	8JUL1145	9.90
I-4	8JUL1145	7.20

REA as a Business Process Pattern and Value Chain Component

Two of the biggest information technology advances of the 1990s were the initiation of business process reengineering and the advent of ERP (Enterprise Resource Planning) systems (David, McCarthy, and Sommer 2003). In many ways, these technologies accelerated the use of REA modeling because its basic pattern had the same microeconomic foundation and rationale.

Figure 6 illustrates how the REA model maps to the foundational frameworks for these two technologies by defining a *business process* and an *enterprise value chain*. The business process definition is due almost entirely to Hammer and Champy (1993, 35), while the value chain definition is an aggregate of the thinking of Porter (1985), Ijiri (1967), Hergert and Morris (1989), and Geerts and McCarthy (1997). Each business process illustrated has a set of inputs (economic resources given or consumed) and a set of outputs (economic resources taken or acquired). The purpose of the network of business processes as it proceeds from left to right is to assemble the cookie's bundle of value-adding attributes for the final customer. For example, with a larger value chain than can be illustrated here, those cookie attributes might be its recipe-inspired taste, its guaranteed freshness, its advertised status of being wholesome, and its delivery to a customer's front door. Each component of a downstream product's portfolio of value should ultimately be traceable to an upstream business process in the value chain (or alternatively, supply chain).

The revenue cycle bubble at the middle right of Figure 6 should be considered as an abstract description of the data model of Figure 4 and the partial relational database of Figure 5. To assist in seeing this abstraction process, Geerts and McCarthy (1997, 1999) developed an REA shorthand notation for illustrating enterprise value chains as portrayed in Figure 7.

FIGURE 6
Definitions of *Business Process* and *Value Chain*

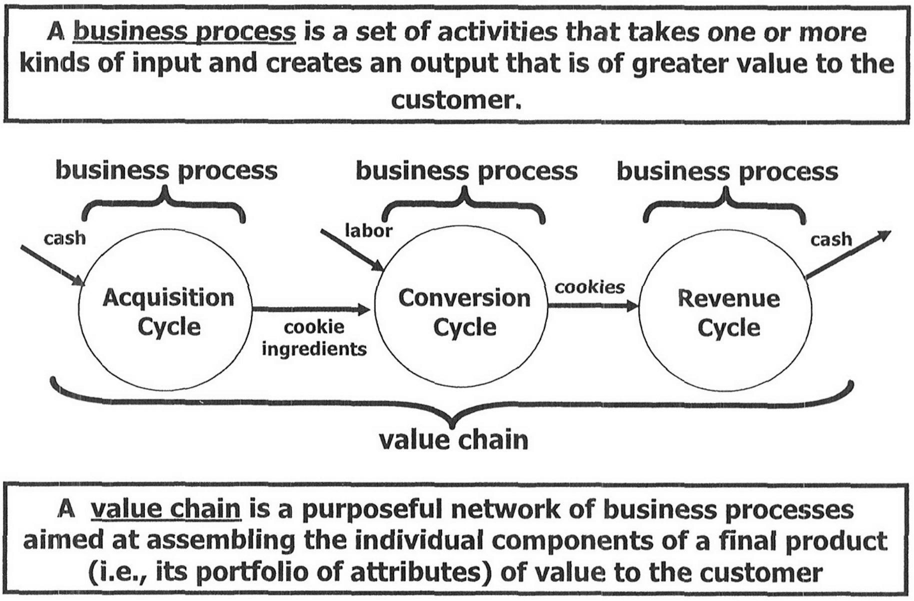
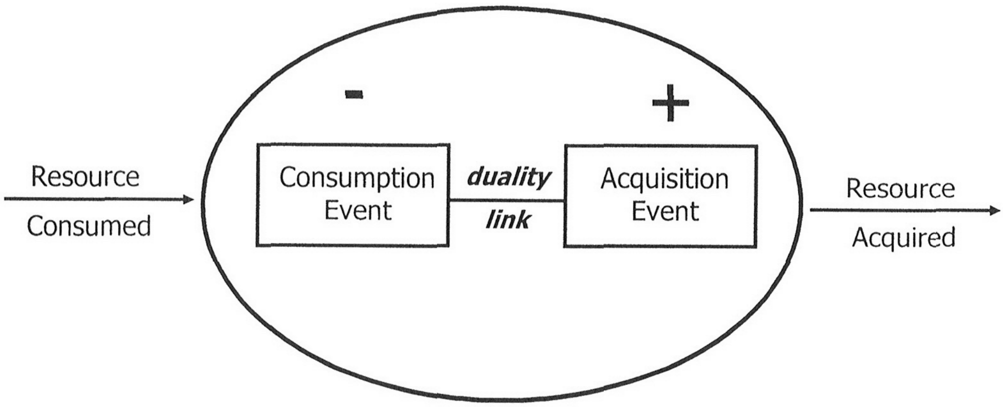


FIGURE 7
REA Shorthand Notation for Enterprise Value Chain



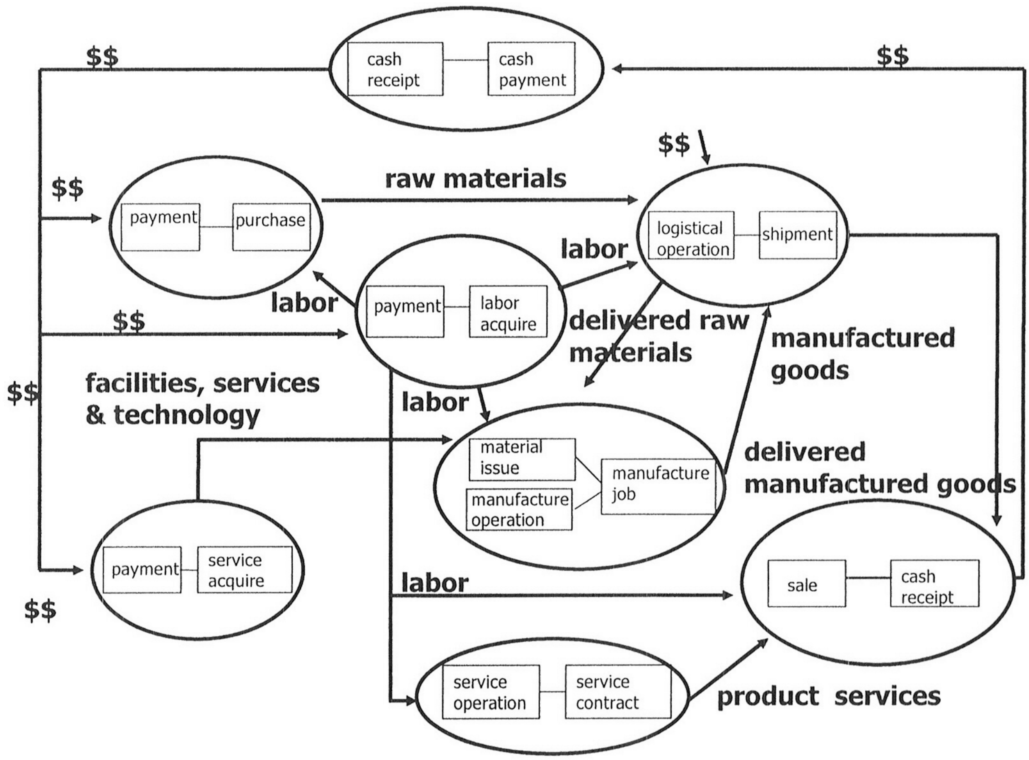
The shorthand notation shows each business process as an oval or circle with input arrows defining resources consumed by the process and output arrows defining resources produced or acquired. Agents are hidden. The major consumption and acquisition economic events are illustrated within bubbles, and as always occurs with any decomposition process, the choice of event granularity to portray is a heuristic decision. These judgments are beyond my scope here, but Geerts and McCarthy (2001) discuss them extensively. The minus and plus signs indicate give and take, but these are sometimes left out where the directional arrows make the value-adding resource flow obvious. When my own students are asked to produce value chains on exams or computer projects (usually a priority task), I ask them to include as much detail as possible within the notational conventions, so I can follow their entrepreneurial logic.

Figure 8 illustrates a generic value chain that I use as an organizing principle at the beginning and end of my junior-level AIS classes at MSU. The financing cycle at the top sprouts the cash flow needed to acquire the factors of production shown as processes on the left. These include facilities, services, and technology plus raw materials (the traditional accounting acquisition cycle) and labor (the traditional accounting payroll cycle). Transportation, manufacturing, and service contracting are portrayed as conversion processes near the middle of the diagram, while the revenue cycle (see Figure 4 and Figure 5) completes the flow at the lower right by cycling cash back to the financing source.

For readers intrigued by the prospect of using REA and value chain construction as part of their accounting instruction, I have some suggestions. For their first few REA courses, most new teachers concentrate on using just the acquisition, payroll, and revenue cycles, because the entrepreneurial logic is most straightforward there, and the representation problems are the least difficult for students to solve. Any kind of conversion process is difficult to understand and teach, so that component goes last in my courses. Detailed treatment of conversion is omitted in many REA-oriented systems classes because of time constraints.

Finally in this section, I need to mention the difficulties associated with the teaching of process decomposition and workflow specification. In AIS terms, this most often means tying the documentation of tasks and business events all together in a value-adding framework with semantic modeling as augmented by tools like system flowcharts, data flow diagrams, and process mapping. Almost all AIS instructors teach such documentation, but very few of us are able to explain comprehensively to students:

FIGURE 8
Generic REA Value Chain



- Why some events merit full representation in an enterprise database while others do not (for example, why include shipment of orders as a relational table, but not picking of orders?); and
- Where the loose temporal aggregation of business events makes economic sense or even where the performance of some tasks is essential from a value-adding perspective (for example, how do we account for the activities of a loan-screening committee for customers?).

There are some very good discussions of these problems in AIS, among which I think Denna et al. (1993), David (1997), and Bradford et al. (2001) are the best. The reengineering literature (Hammer and Champy 1993; Hammer 2001) and some managerial accounting texts contain sound advice as well. Our own best thinking on teaching this material is given in Geerts and McCarthy (2001), but some additional schemes might be resolved soon (Geerts and McCarthy 2003). To me, this is clearly an area that needs additional teaching and research work by the AIS design science community. I do not think that the newer types of cost accounting systems (like ABC) work as well as might be generally considered, because they primarily work from a bottom-up and through-the-general-ledger mindset. Top-down semantic modeling can do better.

Types and Commitments—The REA Ontology

The latest work on semantic modeling of accounting phenomena (Geerts and McCarthy 2002; David et al. 2002) deals with its continuing expansion into areas not usually covered in an integrated way by traditional accounting—that is, the extension of the modeled economic phenomena into types and commitments. Semantic models like Figure 4 have traditionally been limited to accountability concerns (Ijiri 1975) where the focus is on the firm's past set of economic activities, although it is most certainly possible to add nonaccounting attributes and nonaccounting procedures to the modeled environment as well. The inclusion of types and commitments enlarges that accountability focus in very dramatic ways that enable the integrated utilization of the resulting enterprise information systems for a much larger audience of both accountants and nonaccountants. Accountants can move into the automated specification and enforcement of a wider set of control policies, and nonaccountants can use more reliable accountability structures as a base for both building and better managing their marketing, supply chain, and human resource data.

Our best and most current thinking on the use of types and commitments in AIS is contained in the REA ontology work (Geerts and McCarthy 2003). Even though it is tentative, I must admit that this material energizes my teaching, especially when the targeted student audience is one (like M.B.A. students) whose prevailing mindset concerning the wide applicability of accounting information might euphemistically be labeled as "somewhat skeptical." Using types and commitments in AIS teaching is a targeted way to become more integrated. However, as I have already mentioned in the case of conversion processes above, it might also be a teaching direction to approach more cautiously. Teachers new to semantic modeling might better postpone their introduction of such material to a second or third year.

A PLAN FOR TEACHING ACCOUNTING INFORMATION SYSTEMS WITH REA MODELING

A Personal Note on Teaching AIS

When I first started teaching an AIS course in Spring 1978 at MSU, I had no integrated plan, and it did not occur to me that I needed one. I had spent more than three years at UMASS teaching nitty-gritty technology to business students (BASIC and FORTRAN programming plus background MIS technology) and 15 months teaching elementary financial and managerial accounting to sophomores. I thought that AIS teaching would consist simply of throwing these previous preparations into some kind of mix that would eventually work well.

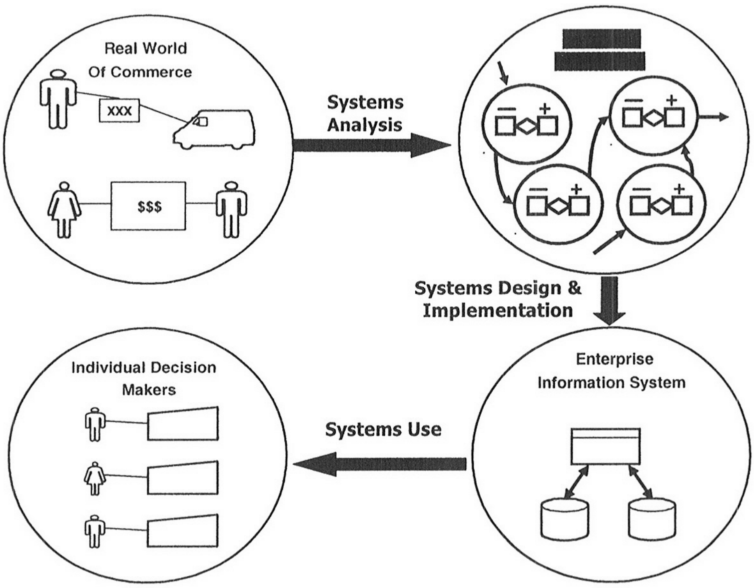
I was dead wrong, and it took me only one term to discover that a potpourri course without any unifying theme was a dreadful experience for both teacher and student. Combining an accounting mindset with the latest technology *du jour* was a mistake I made, and I believe it is a mistake repeated often by AIS teachers, often at the innocent behest of department-wide curriculum committees. Historically, AIS has not had a unifying theme (McCarthy 1999), although there have certainly been some group efforts aimed at producing an AIS teaching charter. Long laundry lists of technical topics make neither a good course nor a sound curriculum.

From my biased viewpoint, semantic modeling of accounting phenomena in general and REA modeling in particular provided that unifying theme. This was an easy choice for me, because I had only to combine my research mindset with my teaching mindset. REA progressed to a larger and larger share of both the main undergraduate AIS class and the larger AIS curriculum at MSU throughout the 1980s, despite the fact that it was exceedingly difficult at times to provide the needed database technology to my students. Research prototypes (Gal and McCarthy 1983, 1986; Denna and McCarthy 1987) were often all I had to work with. This technology deficit largely evaporated in the late 1980s as microcomputer database systems became better and much more common. It was this installation base plus the accumulated expertise of MSU doctoral students in AIS that eventually led to accelerated use of semantic modeling at accounting departments beyond MSU.

The Skill Set for AIS Classes

The best way to understand AIS teaching at MSU is to look at the systems skills involved. These are illustrated in Figure 9 and subsequently explained.

FIGURE 9
Different Skills for AIS Courses



- **Systems analysis** involves investigating the real world of commerce and then producing some kind of a logical framework for the eventual implementation of a system. For a business system, especially one of the ERP class, that logical framework should mirror the firm's own entrepreneurial logic as seen in its value chain. It is my fervent belief that students understand the nature of enterprise-wide logic much better after they have built reasonably complex enterprise models themselves; hence my strong emphasis on the skills that take a student from the northwest corner of Figure 9 (where they perceive processes like the purchase of flour for cash) to its northeast corner (where they construct in their mind an enterprise information architecture for the best possible system for a company doing business that way). Analysis skills for me include being able to outline a complete enterprise value chain and then being able to draw a complete semantic model for each business process.
- **Systems design and implementation** involves adapting a conceptual model to the logical constraints of a particular class of information technology (design) and then making it work on an actual system (implementation). Logical design for me means translating a data model to a set of relations. Implementation means getting that set of tables implemented in both a declarative and procedural sense. These are good skills for students to have, but not as important as analysis skills.
- **System use** involves being able to interact with an existing installed technology, with the ultimate objective of using it to support some decision objective that one might have. This is a needed skill in AIS, but less important than those previously mentioned, especially in our advanced courses. If I am squeezed for time, it is the first thing that goes. For other accounting classes like managerial, tax, and auditing, it is more essential.

Making my students informed users of installed software products is not enough for me. I want them to be able to demand software solutions that transcend an ordinary corporate user's vision. The best way to ensure this is to progress top down from the best solution possible and to then make concessions (known as *implementation compromises*) begrudgingly. This attitude causes students to question openly the adequacy of prepackaged legacy solutions to accounting problems.

THE REA COURSES AT MICHIGAN STATE UNIVERSITY

Basic AIS

Our sole undergraduate AIS class is taught at the junior level; it has no enforced prerequisites. A companion graduate class (where the word "accounting" is replaced by "database" in the title) is taught to first-term accounting master's students and second-year M.B.A.s. The graduate class has a little less accounting, but the exact same REA infrastructure. The skill mix is approximately 60 percent systems analysis, 30 percent systems design and implementation, and 10 percent systems use. The course receives excellent ratings.

Quizzes and Tests

The core feature of the AIS class is a series of conceptual modeling problems that students do as homework, as take-home quizzes, and as in-class examinations. These problems proceed through six levels of difficulty. These levels are enumerated below; beside each description is a data modeling case that tests competency at that level. All of these case are available at my MSU website (McCarthy 2003):

1. Simple business process modeling: *University Slum Lords*
2. Multiple business process modeling with value chains: *Boston Bottle*
3. Complex business processes with type images and policy specification: *Jane's JELL-O*
4. Complex business process modeling over extended value chains: *Outsourced Movers*
5. Complex modeling of conversion processes: *NICOSYS*
6. Complex business process modeling with commitments, types, and contracts: *Vivian's Fashion Factory I and II.*

The problems take from two to three hours to do; in-class examinations are at night. Exams and quizzes count for 40–50 percent of the grade. At present, *Outsourced Movers* is my midterm level, and *Vivian's Fashion Factory* is my second exam level.

These same six problems were used at the second offering of the American Accounting Association's school for beginning AIS teachers (alternatively known as *REA Basic*) at MSU in June 2003. Some students there struggled to finish in six days, but most seemed to make it. As I have mentioned earlier, this is a very ambitious undergraduate case schedule, and I do believe that most AIS teachers who use REA stop near levels 2 or 3.

Computer Projects

The undergraduate AIS class has three computer projects, all of which are available from my website (McCarthy 2003):[2]

1. *M&M Enterprises:* This is an Access tutorial and a simple relational database like that shown in Figure 5. Students can solve problems on paper first and then on an implemented system.
2. *Sy's Fish:* This is a big data-modeling problem that students must solve from scratch. It is about the same level of difficulty as *Jane's JELL-O.*

---

[2] Julie Smith David authored a significant component of all three projects, while Cheryl Dunn co-authored a paper that accompanies *M&M.*

3. *Ventura Vehicles:* This is a preconstructed enterprise database that students use to materialize a virtual general ledger. There is also a significant re-engineering task and a difficult value chain specification component to *Ventura*.

These computer projects account for 20–30 percent of a student's grade. Syllabi for the graduate and undergraduate versions of this course are available at my website. I do cover a number of other AIS topics including traditional legacy (double-entry driven) implementations. However, our clear focus is on the upper range of enterprise implementation choices as identified by David, McCarthy, and Sommer (2003). None of our computer implementations feature traditional packages.

Advanced AIS classes

There are two advanced REA classes in the AIS curriculum at MSU:

1. An advanced database design class that teaches the full range of semantic methods (aggregation, generalization, classification, and typification) and syntactic methods (dependency-driven normalization). Other topics include relational languages and database design methodologies. This class attacks the hardest possible types of representation and value chain specification problems. It also has an intense laboratory requirement. The skill mix shifts to 45 percent analysis, 45 percent design and implementation, and 10 percent use.
2. An object-oriented analysis and design class that uses Java and spends virtually all of its time in the computer laboratory. By the time students reach this class, their conceptual modeling skills are highly advanced, but their implementation skills lag behind. The technology blend here remedies that imbalance. The Universal Modeling Language (UML) is my notation here. The skill mix shifts to 20 percent analysis, 75 percent design and development, and 5 percent use.

The MSU graduate information systems curriculum also features an ERP implementation class (with J. D. Edwards) that uses REA as a conceptual introductory framework for determining the capabilities of the various ERP modules. For the future, we are planning a new XML-oriented class that will use both transaction-level XML (with REA alignment) and reporting-level XML (like XBRL) together. That class will have the three courses mentioned above as prerequisites, so the technology standard for students will be very high.

CONCLUSION

This paper ranges over a number of topics in accounting, in computer science, and in their hybrid field of accounting information systems. Along the way, I have given many recommendations for AIS teaching with an REA orientation. At my MSU website (McCarthy 2003), I keep references and links to a number of REA topics, including the following:

1. *REA research papers*: These include the seminal conceptual papers and much of the 1980s' implementation work, along with many of my latest working papers.
2. *Information and links about textbooks*: I have deliberately omitted mention of specific textbooks here for a number of reasons; primarily because their coverage varies and some of their terminology differs from what I use. On the website, I give very general comments and links to more information.
3. *Information about REA cases and projects used at MSU*: This component was mentioned in the paper's previous section.
4. *Information about REA cases and projects used at other schools*: These links are maintained with the permission of the authors of those materials.
5. *Information and documents describing REA use in e-commerce standards*: This work is especially volatile with versions changing quickly. I maintain links to the latest open material.

Readers with additional links are encouraged to send them to me.

At the beginning of this paper, I modified a quote from Abrial (1974) to fit my own conception of what the REA model aims for in its ultimate implementation:

> An accounting database is a model of the reality surrounding an evolving business enterprise, including its past set of accountability transactions, its present set of commitments and claims, and its future set of plans and policies.

REA research (David et al. 2002) has not yet finished designing all the components of this vision of accounting systems of the future, but I certainly do believe that we have assembled enough to give AIS courses and AIS instructors a strong central theme. As a field, I think accounting in general and AIS in particular concentrate far too heavily on a vision of "what is" and how we can react to that. REA gives an optimistic vision of the best that an implemented system can hope for, and this is what accounting and business students should see in their classrooms.

REFERENCES

Abrial, J. R. 1974. Data semantics. In *Data Base Management*, edited by J. W. Klimbie, and K. L. Koffeman, 1–60. Amsterdam, The Netherlands: North Holland.

Bradford, M., D. Roberts, and G. Stroupe. 2001. Integrating process mapping into the AIS student's toolset. *Review of Business Information Systems* (Fall 2001): 61–67.

Chen, P. P. 1976. The entity-relationship model—Toward a unified view of data. *ACM Transactions on Database Systems* (March): 9–36.

Coad, P. 1995. *Object Models: Strategies, Patterns, and Applications.* Englewood Cliffs, NJ: Prentice Hall.

Codd, E. F. 1970. A relational model of data for large shared data banks. *Communications of the ACM* (June): 377–387.

———. 1972a. Further normalization of the data base relational model. In *Data Base Systems*, edited by R. Rustin, 33–64. Englewood Cliffs, NJ: Prentice Hall.

———. 1972b. Relational completeness of data base sublanguages. In *Data Base Systems*, edited by R. Rustin, 65–98. Englewood Cliffs, NJ: Prentice Hall.

David, J. S. 1997. Three "events" that define an REA approach to systems analysis, design, and implementation. In *Proceedings of the Annual Meeting of the American Accounting Association*, Dallas, TX.

———, G. Geerts, and W. E. McCarthy. 2001. Materialization of account balances in the REA accounting model. Paper presented to the 2001 XBRL Conference, Lawrence, KS.

———, G. Gerard, and W. E. McCarthy. 2002. Design science: An REA perspective on the future of AIS. In *Researching Accounting as an Information Systems Discipline*, edited by V. Arnold, and S. Sutton, 35–63. Sarasota, FL: American Accounting Association.

———, H. Maccracken, and P. M. Reckers. 2003. Integrating technology and business process analysis into introductory accounting courses. *Issues in Accounting Education* 18 (November): 417–425.

———, and W. E. McCarthy. 2003. Ventura vehicles. Case summary, Michigan State University.

———, ———, and B. Sommer. 2003. Agility: The key to survival of the fittest in the software market. *Communications of the ACM* (May): 65–69.

Denna, E. L., and W. E. McCarthy. 1987. An events accounting foundation for DSS implementation. In *Decision Support Systems: Theory and Application*, edited by C. W. Holsapple, and A. B. Whinston, 239–63. Berlin, Germany: Springer-Verlag.

———, J. O. Cherrington, D. P. Andros, and A. S. Hollander. 1993. *Event-Driven Business Solutions.* Homewood, IL: Business One Irwin.

Dunn, C. L., and W. E. McCarthy. 1997. The REA accounting model: Intellectual heritage and prospects for progress. *Journal of Information Systems* (Spring): 31–51.

Everest, G. C., and R. Weber. 1977. A relational approach to accounting models. *The Accounting Review* (April): 340–359.

Fowler, M. 1997. *Analysis Patterns.* Reading, MA: Addison-Wesley.

Gal, G., and W. E. McCarthy. 1983. Declarative and procedural features of a CODASYL accounting system. In *Entity-Relationship Approach to Information Modeling and Analysis*, edited by P. Chen, 197–213. Amsterdam, The Netherlands: North-Holland.

———, and ———. 1986. Operation of a relational accounting system. *Advances in Accounting* 3: 83–112.

Geerts, G., and W. E. McCarthy. 1997. Modeling business enterprises as value-added process hierarchies with resource-event-agent object templates. In *Business Object Design and Implementation*, edited by J. Sutherland, and D. Patel, 94–113. Berlin and Heidelberg, Germany: Springer-Verlag.

———, and ———. 1999. An accounting object infrastructure for knowledge-based enterprise models. *IEEE Intelligent Systems and Their Applications* (July/August 1999): 89–94.

———, and ———. 2001. Using object templates from the REA accounting model to engineer business processes and tasks. *Review of Business Information Systems* (Fall): 89–108.

———, and ———. 2002. An ontological analysis of the primitives of the extended-REA enterprise information architecture. *The International Journal of Accounting Information Systems* 3: 1–16.

———, and ———. 2003. The ontological foundation of REA enterprise information systems. Working paper, Michigan State University.

Hammer, M., and J. Champy. 1993. *Reengineering the Corporation*. New York, NY: Harper Business.

———. 2001. *The Agenda: What Every Business Must Do to Dominate the Decade*. New York, NY: Crown Business.

Hay, D. C. 1996. *Data Model Patterns: Conventions of Thought*. New York, NY: Dorsett House Publishing.

Hergert, M., and D. Morris.1989. Accounting data for value chain analysis. *Strategic Management Journal* 10: 175–188.

Ijiri, Y. 1967. *The Foundations of Accounting Measurement: A Mathematical, Economic, and Behavioral Inquiry*. Englewood Cliffs, NJ: Prentice Hall.

———. 1975. *Theory of Accounting Measurement*. Sarasota, FL: American Accounting Association.

McCarthy, W. E. 1978. A relational model for events-based accounting systems. Doctoral dissertation, University of Massachusetts.

———. 1979. An entity-relationship view of accounting models. *The Accounting Review* (October): 667–686.

———. 1980. Construction and use of integrated accounting systems with entity-relationship modeling. In *Entity-Relationship Approach to Systems Analysis and Design*, edited by P. Chen, 625–637. Amsterdam, The Netherlands: North Holland Publishing Company.

———. 1982. The REA accounting model: A generalized framework for accounting systems in a shared data environment. *The Accounting Review* (July): 554–578.

———. 1999. Semantic modeling in accounting education, practice, and research: Some progress and impediments. In *Conceptual Modeling: Current Issues and Future Directions*, edited by P. P. Chen, J. Akoka, H. Kangassalo, and B. Thalheim, 144–153. Berlin and Heidelberg, Germany: Springer Verlag.

———. 2003. Personal website. Available at: http://www.msu.edu/user/mccarth4/.

Porter, M. E. 1985. *Competitive Advantage: Creating and Sustaining Superior Performance*. New York, NY: Free Press.

Smith, J. M., and D. C. P. Smith. 1977. Database abstractions: Aggregation and generalization. *ACM Transactions on Database Systems* (June): 105–133.

Trimmer, K., C. LeRouge, and R. Pinsker. 2002. Crossing disciplines for educational concepts: The value of REA as a top-down approach to introduce entity-relationship diagramming in IS curricula. In *Decision Sciences Institute 2002 Annual Meeting Proceedings*, 844–849.