Design Science: Building the Future of AIS

by

Julie Smith David

Gregory J. Gerard

William E. McCarthy

Design Science: Building the Future of AIS

This chapter argues that design science is a crucial aspect of accounting information system (AIS) research. Unlike positive research that examines the current state of practice to understand it better, design science strives to identify the means to improve upon it. Thus, researchers using this methodology often "build" new systems to evaluate whether their prescriptions are feasible and to gain deeper insights into the problem being investigated. This type of research is widely accepted in colleges of engineering, and we believe accountants can learn much for our engineering and computer science colleagues.

Although design science has not been widely used in accounting research during the past twenty years, there are some domains that have been enriched by this methodology, such as database accounting systems, expert systems, and object-oriented systems. Because we are most familiar with the database accounting systems work, specifically the Resources-Events-Agents (REA) paradigm, we will use this body of literature to illustrate design science topics.

In the three main sections of the chapter we (1) provide a context for understanding design science, (2) take a historical perspective and highlight significant REA design papers and implications, and (3) propose future research directions in REA design science. We will summarize our findings in the conclusion.

An Introduction to Design Science Research

AIS researchers: Are we social scientists or computer scientists?

Accounting information systems research covers a wide range of diverse topics and methodologies. A number of researchers conduct experimental and field research, evaluating theories, testing hypotheses, and performing statistical analysis. These researchers would be considered social scientists, and they would identify with the terms in the left column of Table 1. The methods and mores of "mainstream" accounting certainly favor this type of research. Yet another important group of researchers emphasize information system construction and software engineering. These researchers would be considered more similar to computer scientists, and they would identify with the terms in the right column of Table 1. As we argue throughout this chapter, both groups of scholars create knowledge and engage in empirical activities. Both groups are needed to advance AIS research -- in fact, there are synergies between the two. So, are AIS researchers social scientists or computer scientists? We believe the answer is "both."

--- Insert Table 1 approximately here ---

What is Design Science?

The concept of design science was introduced by Simon (1969) in *The Sciences of the Artificial*. His thesis (Simon 1996, Chapter 1)[1] is that it is possible to create a science of the artificial (i.e., human-made) as an analog to natural science, hence the term "design science." According to Simon, natural science is concerned with the state of natural things, how they are and how they work. The typical home for such scientists is the

[1] From this point on we will refer to Simon's most recent (3rd) edition of *The Sciences of the Artificial* published in 1996.

university's college of science, but the natural scientists' methods have proliferated throughout other colleges such as the college of business. By comparison, colleges of engineering have been created to address artificial phenomena and teach the design and construction of artifacts that meet desired properties and goals (Simon 1996, 111).

A science of design has important ramifications for professional schools including business. Simon (1996, 111) states:

> Everyone designs who devises courses of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or a social welfare policy for a state. Design, so construed, is the core of all professional training; it is the principal mark that distinguishes the professions from the sciences. Schools of engineering, as well as schools of architecture, business, education, law, and medicine, are all centrally concerned with the process of design.

Simon then points out the irony that "in this century the natural sciences almost drove the sciences of the artificial from professional school curricula, a development that peaked about two or three decades after the Second World War" (Simon 1996, 111). He attributes this phenomenon to the general university culture and the quest for respect professional schools sought (the assumption being that natural science methodologies are more rigorous).

Although some disciplines, such as computer science, engineering, architecture, and medicine have recently returned to design science (in varying degrees), business schools in general have maintained a natural science emphasis since the 1960s. Business school disciplines such as information systems (IS) or information technology (IT) have been caught in the middle of these two sciences. In fact, these alternative views motivated March and Smith (1995) to create a framework for IT researchers. March and

Smith (1995, 252) recognize the importance of both types of scientific activities and the tension between the two types of researchers:

> There are two kinds of scientific interest in IT, descriptive and prescriptive. Descriptive research aims at understanding the nature of IT… Prescriptive research aims at improving IT performance… Though not intrinsically harmful, this division of interests has created a dichotomy among IT researchers and disagreement over what constitutes legitimate scientific research in the field.

Descriptive research and prescriptive research correspond to natural science and design science respectively. Interestingly, Simon (1995, 96-8) points out a similar division of interests in the field of artificial intelligence, which he refers to as the "social fragmentation of AI." In accounting, prescriptive research has for the most part been abandoned (Mattessich 1995). Furthermore, if we examine the recent trend in business school doctoral programs (specifically in accounting and, to some extent, management information systems), it becomes apparent that the overwhelming majority of students are not exposed to design science. However, the merits of natural science versus design science should not be an “either-or” proposition in the academic community.

The March and Smith (1995) Framework

Rather than argue over what constitutes legitimate scientific research, March and Smith (1995, 251) state that "both design science and natural science activities are needed to insure that IT research is both relevant and effective." Given that both activities are necessary, March and Smith create a framework (see Table 2) that encompasses these *research activities* and their interactions with specific *outputs of research*. The design science research activities consist of building and evaluating IT artifacts. The natural science research activities consist of theorizing and justifying how and why the IT artifact works (or does not work) in its environment. The IT research outputs consist of constructs, models, methods, and instantiations. The definition of these outputs is discussed next.

--- Insert Table 2 approximately here ---

According to March and Smith (1995, 256) "Constructs or concepts form the vocabulary of the domain. They constitute a conceptualization used to describe problems within the domain and to specify their solutions. They form the specialized language and shared knowledge of a discipline or sub-discipline." The value of a clearly defined set of constructs is apparent since all scientists are concerned with precision. The evaluation of these, or any, constructs is essentially based on utility. This is because a construct or definition "can be neither true nor false -- i.e., it is not a factual proposition. A definition is simply an explicit statement or resolution; it is a contention or an agreement that a given term will refer to a specific object" (Lastrucci 1963, 77). In other words, a definition is what the writer says it is. However, construct utility is tested over time. New constructs may be introduced and "compete" with the older constructs; presumably, the more useful constructs will persist and the less useful ones will languish.

March and Smith (1995, 256) describe a model as "a set of propositions or statements expressing relationships among constructs. In design activities, models represent situations as problem and solution statements." The term method is used by March and Smith (1995, 257) as "a set of steps (an algorithm or guideline) used to perform a task. Methods are based on a set of underlying constructs (language) and a representation (model) of the solution space. … Although they may not be explicitly articulated, representations of tasks and results are intrinsic to methods. Methods can be tied to particular models in that the steps take parts of the model as input. Further, methods are often used to translate from one model or representation to another in the course of solving a problem."

March and Smith (1995, 258) define an instantiation as "the realization of an artifact in its environment… Instantiations operationalize constructs, models, and methods. However, an instantiation may actually precede the complete articulation of its underlying constructs, models, and methods. That is, an IT system may be instantiated out of necessity, using intuition and experience."

To make these categories of research outputs more concrete we apply them to a database example from the computer science literature. Some important *constructs* in the relational model (Codd 1970) are relations, tuples, attributes, and domains. A table in a database is a relation. For example, a table (flat record) of customers is a relation. A tuple corresponds to a row in a relational table, such as the representation of a specific customer. An attribute is a column in a table that represents one dimension of the table's subject; in the customer table the customer name would be an attribute. A domain is a set of values that cannot be further decomposed such as the set of all customer telephone numbers. Continuing our example, the *model* of interest is the relational model, a logical model that eliminates redundant data. Some *methods* used in conjunction with the relational model are inference rules for functional dependencies, and normalization. One of the earliest *instantiations* of the relational model was developed by IBM Research called System R. In addition System R was the first instantiation of SEQUEL, which later became SQL (Elmasri and Navathe 1994, 185; for an interesting discussion of System R see http://www.mcjones.org/System_R/).

The categories of research outputs in the framework are not mutually exclusive. In other words, since constructs are a domain vocabulary, then the models (the relational model), methods (inference rules for functional dependencies and normalization), and instantiations (System R) within a particular domain would also be considered constructs. The dependence between categories is also apparent since constructs, models, and methods can become operationalized in instantiations. Therefore, scholars may not unanimously agree with attempts to classify research into different cells of the framework, and a specific research project could be classified across many cells.

In spite of this admonishment, later in this chapter we make an effort to "position" REA research papers in the March and Smith framework in order to provide a global view of REA design science research. In the next section, we examine the notion of design science as an empirical endeavor.

Is building a system an empirical activity?

To a person trained in a business school focusing on natural science methods, the notion of computer science or software engineering as an empirical activity may seem foreign, but it is worth consideration. In 1975 the Association for Computing Machinery presented their Turing Award to Allen Newell and Herbert Simon for their work in artificial intelligence, cognitive psychology, and list processing. In their famous award lecture Newell and Simon (1976, 114) persuasively argued, and it is worth quoting here, that computer science is empirical:

> Computer science is an empirical discipline. We would have called it an experimental science, but like astronomy, economics, and geology, some of its unique forms of observation and experience do not fit a narrow stereotype of the experimental method. None the less, they are experiments. Each new machine that is built is an experiment. Actually constructing the machine poses a question

> to nature; and we listen for the answer by observing the machine in operation and analyzing it by all analytical and measurement means available. Each new program that is built is an experiment. It poses a question to nature, and its behavior offers clues to an answer. Neither machines nor programs are black boxes; they are artifacts that have been designed, both hardware and software, and we can open them up and look inside. We can relate their structure to their behavior and draw many lessons from a single experiment…We build computers and programs for many reasons. We build them to serve society and as tools for carrying out the economic tasks of society. But as basic scientists we build machines and programs as a way of discovering new phenomena and analyzing phenomena we already know about. Society often becomes confused about this, believing that computers and programs are to be constructed only for the economic use that can be made of them (or as intermediate items in a developmental sequence leading to such use). It needs to understand that the phenomena surrounding computers are deep and obscure, requiring much experimentation to assess their nature. It needs to understand that, as in any science, the gains that accrue from such experimentation and understanding pay off in the permanent acquisition of new techniques; and that it is these techniques that will create the instruments to help society in achieving its goals.

Although it seems that many natural scientists do not regard design science as empirical, Newell and Simon offer a different perspective. Ultimately, design science activities are building programs or systems to perform experiments. We caution, however, that although computer science is an empirical activity, that does not necessarily qualify it as research in the academic sense. We elaborate this point in the next section.

Differentiating Between Research and Development

Because accounting academics receive training in natural science methods in their doctoral programs, most can evaluate whether such papers contribute to the literature. Since there is less training in design science techniques, many researchers are unable to confidently differentiate between simple development, and truly academic research projects. In an attempt to provide guidance during a volatile (in terms of quality) period of expert systems research in the middle-to-late 1980s, McCarthy, Denna, Gal, and

Rockwell (1992) developed a framework to assess contributions as either research or development or both. We build on this framework and suggest the following criteria.

Is the research truly novel, given the current state of the field? This question implies that early in a field's development, relatively simple system designs and proof of concept implementations are valuable research activities. However, as a field matures, researchers must move beyond the "Build" column in the March and Smith framework and "Evaluate" their work compared to studies that preceded it. Making only minor design changes, or implementing the same elements with a new tool, are development activities rather than research.

Is the problem being addressed a "difficult" or "easy" one? It is obviously preferable to study challenging aspects of a problem rather than focusing on its simple parts. Therefore, before beginning new projects, we recommend that researchers garner extensive domain knowledge and divide the problem into components or modules. Once segmented, researchers should select the most complex modules to explore, contributing the most to the literature. Of course, if even the most complex module is easy to solve because others have already done it, then future work with the problem will not be acceptable as research. Sometimes, however, a problem is so difficult and situation specific that the researcher's insights will be costly to achieve and not generalizable. In these cases, we believe that commercial firms with large R&D budgets and financial incentives are better suited to resolve the problem. Therefore, the researcher must strike a delicate balance on the easy—difficult continuum.

Having said this, we must recognize that a valid scholarly activity is evaluating a class of problems and abstracting their common characteristics to simplify the problem.

For example, one AI system, GPS, was developed to study task-independent components of decision-making (Ernst and Newell 1969 as discussed in Simon 1995). Thus, the researchers had to identify fundamental components that spanned decision-making domains, and they evaluated their system in over a dozen situations. This definitely constituted research!

Is there already a proof of concept or of feasibility? This question has several implications for researchers. First, when a new design is proposed, implementing it to prove its feasibility is scholarly research. However, if someone else has already developed a similar system, using a new programming language or tool set is a development activity unless the new environment sheds new insights on the research question. Before a work is considered research, the author must take a responsibility to highlight the contributions showing why the new implementation has increased knowledge.

Second, if a study is extending an existing model, the extensions should be implemented as proof of concept. It is important that the new model performs significantly differently than the previous, and, ideally, the analysis should highlight how management's decisions would improve with the new system. Thus, once the research community-at-large accepts a particular instantiation, the onus is on future researchers to prove the superiority of their proposed solutions. The only way this can be done is with an instantiated system.

As a final method of differentiating between research and development, we suggest reading contributions to the literature that have been identified as outstanding design science scholarship. As an exemplar we recommend Codd's (1970) "A Relational

Model for Large Shared Data Banks" – winner of the 1981 ACM Turing Award. In this seminal paper Codd proposed the details of a model based on the mathematical concept of "relation" that separated logical aspects from physical (implementation) details. At this point in time it may seem difficult to imagine not separating the logical from the physical, but this was clearly an insightful and novel contribution. Furthermore, this work facilitated massive new efforts in the areas of database design and procedural specification. Codd was definitely working on the difficult, rather than easy, problems. Additionally, the instantiation of his model were proven better than prior instantiations on a number of dimensions. Later, we will return to our discussion of Codd (1970) to show how this work influenced REA design science research.

Design Science Summary

We close this section of the chapter with recognition that there is no one perfect research methodology, and we call for unity in the AIS field. The prevailing view in both the fields of information technology and accounting is based on positive theorists, mainly in the tradition of Popper. But design scientists subscribe to a different philosophy and this can cause a schism in the research community. However, it is worth noting that even popular methodologies are open to question. Earman, a philosopher of science, argues:

> The philosophy of science is littered with methodologies, the best known of which are associated with the names of Popper, Kuhn, Lakatos, and Laudan…I have two complaints. The first stems from the fact that each of these methodologies seizes upon one or another feature of scientific activity and tries to promote it as the centerpiece of an account of what is distinctive about the scientific enterprise. The result in each case is a picture that accurately mirrors some important facets of science but only at the expense of overall distortion. The second common complaint is that these philosophers, as well as many of their

> critics, are engaged in a snark hunt[2] in trying to find The Methodology of Science (1992, 203-4).

Similar acknowledgements have been published in the accounting literature (e.g., see Hines' 1988 *The Accounting Review* article).

The most definitive defense for including both positive (natural science) and normative (design science) in a concentrated attack on practical accounting problems has been raised by the senior accounting scholar Richard Mattessich in his 1995 treatise *Critique of Accounting*.

> Academic accounting – like engineering, medicine, law, and so on – is obliged to provide a range of tools for practitioners to choose from, depending on preconceived and actual needs. … The present gap between practice and academia is bound to grow as an increasing number of academics are being absorbed in either the modeling of highly simplified (and thus unrealistic) situations or the testing of empirical hypotheses (most of which are not even of instrumental nature). Both of these tasks are legitimate academic concerns, and this book must not be misinterpreted as opposing these efforts. What must be opposed is the one-sideness of this academic concern and, even more so, the intolerance of the positive accounting theorists toward attempts of incorporating norms (objectives) into the theoretical accounting framework (183).

Although he is not intimately familiar with the field of computer science, Mattessich is a strong and vocal proponent for the type of normative endeavors embodied in design science as defined by March and Smith. He even intimates that he is humbled as an accounting academic when he compares "the scientific contributions of accounting – as impressive as its "input" may have been during the last few decades – with the actual results in the natural sciences or such applied sciences as medicine and engineering" (1995, xviii). Again, we agree with Mattessich, and with March and Smith, in their

[2] This is a reference to Lewis Carroll's (1876) poem *The Hunting of the Snark: an Agony in Eight Fits*. It can be found at http://etext.lib.virginia.edu/britpo.html (McGann and Seaman n.d./2000).

opinions that neither normative nor positive researchers in accounting should try to trump the other camp. What is most apparent is that in recent years "we [accounting academics] have *not* done enough to serve the practitioner, the stockholder, and above all, society at large" (Mattessich 1995, 209). A prime contention of this paper is that an influx of design science work in AIS is a way to close this "contribution gap."

We argue that it useful for researchers to draw from the many aspects of science, including design science, to guide our endeavors and enable us to organize our thoughts and knowledge. However, we should not unilaterally adopt one chosen Methodology of Science to the exclusion of all others. Regardless of whether we adopt a design science or a natural science perspective, the issue of primary importance is our motivation for pursing a particular research project. In other words, is the research question interesting and relevant? Does each project make a significant contribution?

The REA Model as an Example of Design Science Development

Introduction

In this section of the paper, we will use the notion of design science with its accompanying set of constructs as developed in the previous section of the paper to explore the initial specification and the attendant development of the REA accounting model. Our treatment here will focus on the research output categories of design science developed by March and Smith: *constructs, models, methods*, and *instantiations*. Readers will notice that our exemplars concentrate heavily on the REA model work done at Michigan State University (MSU). There are two reasons for that. The first is that REA originated there and a good deal of the follow on research (especially in design science) has come from researchers at MSU. The second is that this corpus is the best known to the authors of this paper. Furthermore, an analyst who tries to trace the origin of AIS design work, while its major components have flowed back and forth from reference disciplines (like computer science), needs to understand how the ideas actually developed from origin to final publication.

We believe this review of REA work constitutes a well-developed example of design science in AIS. The major lesson we hope to impart in this review is the following. The invention or creation of new constructs or models for accounting systems can be done in isolation where the individual researchers assess the status quo of accounting practice and then make specific recommendations for improvement. More probable, however, is the scenario where advances in a cognate discipline have been proposed independently, and an accounting researcher then takes that advance and affords it the domain specificity of applied accounting (O'Leary 1988). Hopefully, this cross-fertilization then rebounds back across disciplinary boundaries where the insights developed from the accounting context give the cognate discipline more insight into further developments. With this purpose in mind, we have developed this section with three major tables.

1. Table 3 illustrates design science papers or books that have had major influences on REA development. Most of these papers have decided origins in computer science, and in fact, the list of authors shown includes two winners of the ACM Turing Award -- Ted Codd and John McCarthy -- the highest honor accorded researchers in that field. It also includes three papers (Codd, Chen, and Lum et al.) plus two books (Porter and Gamma et al.) that are considered to be the seminal pieces in the development of major normative areas of research and practice:

relational databases, semantic database modeling, database design methodology, enterprise value chain specification, and design patterns. Readers should note that we have omitted normative accounting theories like those of Ijiri (1975) from this list as those origins have been reviewed in detail elsewhere in previous publications (e.g., Dunn and McCarthy 1997).

2. Table 4 illustrates some major papers that have made significant design science advances in the more focused area of the REA model. The list contains work that exemplifies all four of the March and Smith categories of constructs, models, methods, and instantiations. Readers should note the heavy correspondence of Table 3 with Table 4 (although there is certainly not an even remote approximation to a one-to-one mapping). In a very general sense, Table 3 illustrates the more general pioneers with Table 4 detailing how those more general ideas were adapted to business enterprises most generally and to accounting more specifically.
3. Table 5 is more inclusive and more specific than Table 4, and it is organized not around individual papers, but around the familiar theme of categories of design science contribution. This table has two purposes. First, it gives more specific examples of the types of advances outlined more generally in Table 3. And second, it gives a novice researcher in either AIS generally, or REA more specifically, a place to start their explorations of this field.

--- Insert Tables 3, 4, and 5 approximately here ---

In the three sections that follow, we use the tables defined above as foundations. We follow that with a summary that concludes this portion of the paper.

The Seminal and Definitive Origins of Cognate Research Work that Affected REA

There certainly have been many major advances in computer science since its origins nearly a half-century ago, but we think the most important to accounting systems (in terms of both chronology and overall importance) has been the development of ideas in database theory. Major advances in the 1970s were followed by an integration with the fields of artificial intelligence in the 1980s under the general heading of knowledge-based systems, and later with the field of software engineering under the general heading of object-oriented programming, languages, and systems.

Database Theory. This field had many notable pioneers in the 1960s (like Charles Bachman, the originator of the navigational network model), but its defining moment was the development of the relational model by Codd during the period of 1969-1972. This is an area that was discussed as an exemplar previously in this paper, and it was a field that was fortuitously synchronized with the developing need in accounting systems for a technology platform that would allow a database orientation (as defined by Dunn and McCarthy in 1997):

1. data must be stored at its most primitive levels (at least for some period),
2. data must be stored such that all authorized decision makers have access to it, and
3. data must be stored such that it may be retrieved in various formats as needed for different purposes.

Noticing this symbiotic relationship between accounting systems and database theories is an insight often credited to George Sorter (1969), but it was in fact Colantoni, Manes, and Whinston (1971) -- the second work of Figure 3 -- who first explored its synergy. Their synthesis was based on pre-Codd database technology, and it was left to

Everest and Weber seven years later in 1977 to fully explore the effects of constructs like normalization on traditional accounting structures such as double-entry ledgers. In the meantime, the field of semantic data modeling had emerged to lend more "meaning" to Codd's original constructs with (1) the seminal work of Peter Chen (1976) on the abstraction mechanisms of classification and aggregation, and (2) the follow-on work of Diane and John Smith (1977) on generalization abstractions. Somewhat concurrently with these semantic advances, efforts were being pursued on working relational prototypes with attendant specificational language features, the most notable of which was the System R project at IBM in San Jose which pioneered the development of the SEQUEL (SQL) language (Chamberlin et al. 1976). These declarative and procedural database foundations were made further applicable to event-oriented fields like accounting by Bubenko (1977) who explored the very important ramifications of updating stock entities (like inventory) over time intervals with flow events (like purchases and sales). This was a phenomenon he investigated under the general rubric of "conclusion materialization." The entire field of both syntactic and semantic design of database systems was summarized and categorized in the definitive textbook of Tsichritzis and Lochovsky in 1982 wherein they gave precise definitions to ill-defined and often misunderstood notions such as the difference between specificational (set-oriented) and navigational (element-by-element) languages. And finally with respect to databases and their application to business enterprises and accounting, the work of *The New Orleans Database Design Workshop* (Lum et al. 1979) emerges as particularly significant. Prior research work had concentrated inordinately on "toy" problems with just 4-5 relational objects, and this workshop changed that with the publication of a methodology that:

(a) separated conceptual, logical, and physical database design, and

(b) further called for controlling the complexity inherent in large-scale enterprise applications by separating and sequencing the solution of small local database problems (view modeling) with their integration to a global schema (view integration).

Knowledge-Based Systems and Object-Orientation. All of this computer science and database accounting work had set the stage for the emergence of semantic models of accounting phenomena like REA in 1982. These 1970 advances in the field of database theory were followed by consolidations during the 1980s and 1990s with the fields of artificial intelligence and object-oriented representation. In our estimation, the best way to understand this amalgamation of the last 20 years is to study carefully the definitive texts of John Sowa. While actually being published in 1984 and in 1999, Sowa's books were really compiled and written throughout the decade prior to each release. They integrate well the richer context and capabilities of knowledge-based systems and their cognate disciplines of psychology, linguistics, and philosophy, and they make specific distinctions that later proved to be important to REA development like conceptual relativity and the primacy of declarative representation. To these background frameworks, we add to Table 3 two specific publications that caused changes in REA thinking, one a research paper and the other a software engineering book. The first of these contains an idea generally credited to John McCarthy that he called *epistemological adequacy*, a notion that created the context for the development of *full-REA* systems in the 1990s. The second of these was a 1995 book by Gamma, Helm, Johnson, and Vlissides that strongly encouraged the development of design patterns as an approach to software engineering, an tactic being explored for REA yet today (Geerts and McCarthy (1997c).

Summary of the Seminal and Definitive Design Science Origins of REA. With the exception of one major work by Michael Porter, we have now reviewed the context of the work in Table 3. Porter (1985) was published as a treatise on strategic management, and one of its components was the formalization of an idea used elsewhere by both management theorists and economists: the enterprise value chain. Porter's conceptualization of a value chain provided the theoretical context for stringing business processes together with resource flows by Geerts and McCarthy in the 1990s. His idea was only a component of a larger strategic framework, and it does differ slightly from the entrepreneurial script of Geerts and McCarthy (1999) in that it allows the notion of *ex ante* specification of support activities, something which they allow only as *ex post* implementation compromises.

We leave this review of major design science publications with two caveats for the reader. The first of these is a reminder that these origins concentrated on contributions that are most familiar to the present authors because of their own experience in the field. The second (and clearly more important) piece of counsel is this: researchers (especially novice researchers) should not automatically assume that **any** major advance in a cognate field like computer science can automatically be imported into a field like accounting systems where it will, without question, be recognized as a research contribution. Some advances in cognate disciplines have no applicability to accounting problems. More problematically, some advances have applicability, but their introduction brings no clear advance over existing proposals and implementations. Important advice here is to consider again the framework of Table 2 and to be able to convince oneself that the new import will produce either a novel construct, method, model, or instantiation (first column *build*) or a design contribution that ranks better on some established research metric (second column *evaluation*).

Some Papers That Have Made Significant Design Science Advances in REA Modeling

Table 4 lists a number of papers that have made what we consider to be significant advances in REA design science. The details of many of these advances have been cataloged under the *construct-model-method-instantiation* taxonomy of March and Smith in Table 5, but the purpose of this section is to describe more generally the overall effect of these published works.

Seminal Exposition. The two *Accounting Review* publications listed in the first two rows of Table 4 obviously constitute the seminal exposition of this model. McCarthy (1980) contains procedural specifications in SEQUEL that were originally included in the 1979 paper, but which were rejected by accounting reviewers as too computer-specific. Those computer science contributions -- which were crafted from a combination of specifications given by Chamberlin et al. (1976) and actual discussions with the System R design team -- were published instead in the proceedings of the first *Entity-Relationship Conference* organized by Peter Chen in 1979. Together, these three papers, in both specific and general fashion, outlined a new set of semantic primitives and an overall model of how those primitives fit together that could be used collectively to specify accounting systems. REA approached the task of accounting system design in an entirely new fashion that obviated many of the difficulties being identified at that time with the adaptation of traditional accounting practices to systems of the computer age. The new REA proposals overreached in the sense that more hospitable implementation environments for many of their proposed changes were not present in the early 1980s. REA models had to await changes in the following categories before their effects could be fully realized:

(1) hardware technology (faster processing speeds, better direct retrieval methods, cheaper storage, and (especially) better source data automation),

(2) software technology (object-orientation with pattern driven analysis and design),

(3) business methods (business process engineering, activity-based costing rationale, and enterprise-wide coordination of resource flows), and

(4) communication environments (e-commerce with its need for consistent inter-enterprise semantics and active ontologies).

Network and Relational Implementations. The third and fourth rows of Table 4 indicate work carried out by Gal and McCarthy at Michigan State University in the early 1980s that strove to implement many of the REA ideas in actual database environments. Both implementations preceded the widespread availability of desktop computing, so they were done on mainframes. However, they both used systems that were later to become successful in PC environments. The network system used *GPLAN* in 1980 and 1981 as it was developed at Purdue University (Haseman and Whinston

1977), and the relational implementation used *Query-By-Example* (QBE) in 1982 as it was developed at IBM Research in Yorktown Heights (Zloof 1975).

The research contributions of both database prototypes were many, primarily under the category of new methods and (somewhat obviously) new instantiations. A network heuristic method learned here was summarized thus by Geerts, McCarthy, and Rockwell (1996):

> It is usually the case that relationships between all classes of the economic resource "inventory" and the economic events that affect it are "many-to-many" and that these relationships thus necessitate a CODASYL intersection record-type to both effect the link and provide a home for any jointly dependent attributes. Furthermore, the procedural uses of this data structure involve most commonly a sequential access path through the more stable "inventory" entity. Therefore, in the E-R to CODASYL translation, provide automatic schema definitions for these facilities whenever this pattern is encountered.

In the relational implementation, Gal and McCarthy materialized the entire accounting trial balance with a single hierarchical set of procedures, work that led subsequently to other relational implementations in more complicated environments (Denna and McCarthy 1987) and to a generalized framework for procedural materialization of all account data (McCarthy 1984). They also encountered some counter-intuitive ideas such as the discoveries that (1) a set-only language like QBE couldn't be used to produce LIFO or FIFO inventory numbers, and (2) null values in sets that did not monetarily equate to $0.00 as one would expect from ordinary accounting discourse.

REA CASE Tools. The fifth and six rows of Table 4 represent efforts in building CASE (computer aided software engineering) tool prototypes for REA. In both cases, the original system architectures were outlined in papers presented at the *Avignon AI and Expert Systems Conference*, while the implementations followed some time later with the publication of results even later still. An overview of the contributions of both tools, along with those of other REA CASE tools, was given by Geerts, McCarthy, and Rockwell (1996).

The REACH prototype was first outlined by McCarthy and Rockwell in 1989, and the REAVIEWS component of that system was implemented in a LISP-based AI system, *GOLDWORKS*, by Rockwell in 1992. REACH developed a number of novel heuristics for view modeling, view integration, and (especially) implementation compromise.

CREASY was a PROLOG-based tool of much smaller scope than REACH, but its main contributions were not of the software engineering heuristic variety. Instead, its development led to some theoretically ambitious metrics for any pattern-based reasoning tool with its embodiment of constructs like *epistemological adequacy* and *intensional reasoning*. CREASY is an outstanding example of a research effort whose original base came from computer science, but whose ultimate development resulted in contributions that rebounded from accounting back to computer science. The CREASY development of pattern-matched procedures in operational use presaged by some years the development of active ontologies in AI (Guarino 1998).

The REA Value Chain Model. The series of papers presented and published by Geerts and McCarthy (1994, 1997a, 1997d, 1999) in the seventh row of Table 4 represent the most significant change to REA since its initial specification 1982. The original REA pattern dealt with single exchanges, although the concept that all resources must have both inputs and outputs modeled provided a method to string exchanges or processes together. Geerts (1993) formalized this idea with the notion of a *scenario*, and he and

McCarthy applied the enterprise-wide extension of this concept to the Michael Porter notion of value chains in 1994. Geerts and McCarthy (1997a, 1997d, 1999) specified the REA value chain model more precisely, and they added the notion of *tasks* (compromised decompositions of business processes). Readers should note that the ideas of *tasks* developed here and the Julie Smith David notions of *business event* and *information event* (described below) are different approaches (developed independently) to the problem of defining very similar types of phenomena.

The Database, Semantic, And Structuring Criteria. The *JIS* paper by Dunn and McCarthy in 1997 was primarily a historical review that tried to assess and reestablish the line of contributions to the ideas of disaggregate and multidimensional accounting systems. In the process of doing that however, they discovered that terms like "events accounting" were ill understood and that differentiating different classes of systems was very difficult in the absence of usable criteria. To remedy this difficulty, they established three progressively finer definitions that they called a database orientation, a semantic orientation, and a structuring orientation. These criteria were then used to catalog research efforts in the wider arena of multidimensional and disaggregate accounting systems.

The REA Ontology. The most recent additions to the REA model (last row of Table 4) were proposed by Geerts and McCarthy (2000b) who expanded the existing set of defined entities and relationships in two major directions with type images and commitment images. This expansion was explained using the notions of the emerging AI field of *domain-specific ontologies*, more specifically using the 12-part ontological categorization scheme of John Sowa (1999). Sowa uses three ways to divide categories:

(a) concrete -- abstract, (b) continuant – occurrent, and (c) firstness – secondness – thirdness. This is a classification scheme based heavily on the philosophical ideas of intellectual giants like Aristotle, Kant, Peirce, and Whitehead (Sowa 1999). This division (a 2x2x3 factoring) gives twelve overall categories that Geerts and McCarthy used to explore the extension of existing REA definitions from an *accountability infrastructure* to a *policy infrastructure*. This initial REA ontology work is presently being extended with more integrated use of microeconomic theories and definitions (Geerts and McCarthy 2000c).

Our review of the Table 4 papers is now complete. In reviewing both Table 3 and Table 4, we remind readers of the cross-fertilization possibilities in design work that flows from computer science to AIS and back again. These are certainly two vibrant and emerging disciplines, and the opportunities to take advances in one and apply to the other should only grow as information technology becomes more pervasive.

Individual Listing of Significant Constructs, Models, Methods, and Instantiations in REA Modeling

Table 3 and Table 4 gave explanatory overviews of computer science and REA design advances. Those explanations concentrated on the paper and book level, although there certainly was sufficient detailed explanation of individual advances. This section of the paper and Table 5 zoom in for a more detailed look at precise definitions of the four March and Smith contribution categories – constructs, models, methods, instantiations -- as they apply to REA. The explanation sections here are shorter because most of the detail is given the table sections.

REA Constructs. The first section of Table 5 lists the published REA constructs and gives a definition or description of each. The first eleven constructs come from McCarthy (1982) who used the abstraction methods of aggregation and generalization to derive the REA primitives. In looking at these definitions, we echo a caution first given in Dunn and McCarthy (1997). Many of the REA (abstraction-derived) primitives resemble ideas of normative theorists like Ijiri (1975), and McCarthy used this resemblance to position his constructs within their normative frameworks. However, users are reminded that the exact definitions and their connections with each other are the ones given in the 1982 paper.

The 1982 paper did not specifically deal with the database area of *constraints*, and this was an area attacked soon thereafter in the referenced paper by Gal and McCarthy (1984).

The initial set of REA constructs is followed by one introduced by Denna, Cherrington, Andros, and Hollander (1993) in a text originally written for practitioners and later expanded to an AIS textbook (Hollander, Denna, and Cherrington 1995). This was the idea of *location,* which they added to give more dimension to the original notion of an economic event. This construct is followed by the REA-specific meaning of *implementation compromise,* which is something explored and discussed extensively by Rockwell and McCarthy in their REACH work. Three primitives of the REA value chain model are given next followed by three ideas from David's (1997) Three Events Model. These in turn are followed by the orientation definitions espoused by Dunn and McCarthy (1997). The most recently-published REA contributions of Geerts and McCarthy (2000a, 2000b, 2000c) spawn the last six construct entries of Table 5.

To this list we could add other constructs, especially others that are presently espoused in working papers, such as *time* (O'Leary 1999b), *ex ante accounting objects* (Verdassdonk 1999; 2000), and *external REA models* (McCarthy 2000). There are also a number of other constructs published that cover developments somewhat similar to the domain of REA -- like the work of Seddon (1996) and Adamson and Dilts (1995) – but we do not include them here because their very traditional approaches (both are double-entry oriented) make them impossible to integrate within our intended purpose here.

REA Models. Under the March and Smith category of design science models, we give five examples. The first of these is the original REA model which ties many of the constructs given above into a comprehensive and cohesive approach to building accounting or enterprise information systems in the types of shared data environments that characterize business enterprises today. The second example model includes all the elements of the first with the additional layers of value chain specification abstracted above and of task specification detailed below. This is followed by David's Three Events model which adds business event, information event, and new relationships like synergy to REA. We then show the theoretical framework from which Geerts and McCarthy derived their value chain abstractions with an illustration of the value chain and value system components of Michael Porter's strategic management model. And finally, we illustrate the newer components of the REA ontology as that set of constructs is being assembled by Geerts and McCarthy.

REA Methods. Methods, according to March and Smith, are guidelines used to perform a task, and the first six entries under this heading include Michigan State work that has been discussed already. The listed work by Nakamura and Johnson comes from the patterns research group at the University of Illinois where the implementors used their expertise in this emerging area to illustrate how inefficient materialization of account balance information in REA models could be facilitated with the use of object-oriented design patterns. The last entry by O'Leary details methods for adapting REA models to data warehouse construction, a problem quite similar to the notion of view materialization.

REA Instantiations. Information technology instantiations are real working systems, albeit often at the prototype level. The last heading of Table 5 includes eight of these with both prototype and production status.

The first five instantiations shown are research prototypes. The initial one of these was done by Armitage under the sponsorship of the Management Accounting Society of Canada, and its purpose was to illustrate how REA-oriented systems could produce managerial decision data in a manner that bettered traditional manufacturing accounting systems. The next three rows represent CASE tool prototypes, two of which – REACH and CREASY – have been explained already. The third was done at the University of Southern California by two computer scientists (Chen and McLeod); the aim of REAtool was to support database evolution of the types specified by Batini, Ceri, and Navathe (1992). The last prototype was actually a user interface developed by Dunn as part of an empirical assessment. Her interface was written in VisualWorks (a SMALLTALK tool), and it supported all the major types of data abstraction: classification, aggregation, and generalization.

The last three rows given in Figure 5 represent production systems that are working in actual practice. The IBM payroll system and all of the Price-Waterhouse

GENEVA systems were implemented by teams that included the REA design expertise of Eric Denna. GENEVA does not represent a single implementation, but a practice unit of Price-Waterhouse that actually implemented multiple REA-type systems at firms like Sears. And finally, the last implementation illustrated represents a supply chain coordination system developed by a firm founded by Robert Haugen and others. Haugen's system uses REA patterns of market exchanges and internal transformation processes in sequenced order to create the models for optimizing dependent demand in multi-firm supply chains.

Summary of the REA Design Science Examples

We have now finished our journey through the design science examples of Tables 3, 4, and 5. Our purpose in conducting this accelerated review was threefold. First, we wished to emphasize that the field of computer science is the arena we consider most fruitful for the germination and the exportation of good research ideas for AIS design scientists. The learning curves here are very steep, especially for scholars trained in traditional accounting doctoral programs, but the rewards are large and very sustainable over a long period of time. To the average lay person, it seems that computer technology progresses in unpredictable ways, and while this is always true to some extent, that progress is actually much more patterned than it looks. Good software engineering, database, and AI foundations molded 15-20 years ago can be used still to produce quality streams of research products today with a surprisingly modest amount of educational maintenance. Second, we wished to illustrate the integrative nature of most of the REA work, especially the REA work conducted at Michigan State. Traditional accounting researchers sometimes view technology projects as one-of-a-kind efforts that have little or no accumulated tradition and direction, and it would certainly seem obvious from this review that this is simply not the case. And our final purpose in this section was to set the stage for the last part of the paper that takes the review done here as a context on which to expand our vision to the future of design science research in general and REA design projects in particular.

Future Research

Although a solid REA foundation has been developed in previous literature, there is still a wide range of opportunities available to help "build the future" of accounting information systems and to enhance our understanding of the REA model. This section provides an overview of several types of studies we believe can make significant contributions to the literature. First we provide several ideas for extensions to the REA model. These projects would rigorously evaluate REA in more complex environments and further our understanding of both accounting techniques and AIS. Second, we have identified two new areas of REA research that propose analyzing characteristics of commercial software and REA. We discuss coupling the exploration of REA with today's enterprise resource planing systems (ERP). Studying commercially available products provides the opportunity to use design science techniques to evaluate the fundamental REA literature. Additionally, we believe that identifying areas in which the ERP systems differ from the REA pattern may identify opportunities to further the development of enterprise systems. Finally, we describe opportunities to truly challenge the REA model by applying it to today's emerging e-commerce business models. As we will show, the Internet markets of the future may lead to radically different data structures than we see today. As such, systems developers may need to design systems that store data in one place, but allow users from different organizations to view the data according to their frame of reference.

REA Extensions

To date, the examples provided in REA research focus on three major business processes: sales, procurement, and manufacturing. Because these areas have been studied in depth, it will be very difficult to identify new research questions using these cycles[3]. Therefore, the opportunities to extend the REA research require applying the model to more complex situations. Such studies could result in proof of concept implementations of new domains, extensions to the basic REA model, or the identification of situations in which the model is insufficient. They will provide evidence of the model's "generality," one of the key evaluation Simon metrics identified to evaluate artificial intelligence research projects (1995, 103-104), and we believe it applies to this area of design science research, too. The following paragraphs describe potential research questions.

Equity Transactions. McCarthy (1982) includes a brief discussion of how equity transactions should be modeled. Owner's Equity is the sum of a firm's capital stock value and its retained earnings. While retained earnings can be calculated procedurally, the system must track the value of capital stock. To do this, the basic REA template would specify that the amount of capital stock a firm should recognize as the difference between the cash receipts it receives from stock subscriptions and the dividends that have been declared. However, because there are so many attributes associated with the stock subscription and the events associated with it, McCarthy (1982) describes an extension to

[3] This is perhaps not as true for manufacturing which was explored in Armitage (1985), Denna, Jasperson, Fong, and Middleman (1994), and Grabski and Marsh (1994). Because of the complexity of this cycle, additional work may prove valuable.

the REA model to explicitly reify the relationship between the receipts and disbursements (see Figure 1). In this figure, the duality relationship between the receipts and disbursements is portrayed with dotted lines because it would be replaced with the entity set included in the cloud on the right side of the figure. A system created from this model would track details about the Stock Subscriptions and Dividends Declared, in addition to their related cash transactions.

--- Insert Figure 1 approximately here ---

Although this extension would enable the system to calculate Owner's Equity, there are equity transaction details that would not be supported without further extensions. For example, when shares are re-sold in the market, the firm must be able to track the new shareholders for future dividend distributions. Similarly, when a distribution is declared, but not yet made, the system must track shareholders that will receive dividends, versus those who have purchased the shares after the declaration date.

More importantly, there are additional equity transactions that have yet to be explored. How would stock options be recognized in an REA system? Consider how many financial instruments today's financial services firms offer. How can a system be designed that can enable flexibility to offer new products yet track the data so that the firm can consolidate information from across the products? There are also important questions to ask if one is modeling equity transactions from an investor's point of view. For example, how do we model derivatives?

We believe the key to performing this research is a very rich understanding of the equity procedures and REA. By combining the two, researchers would be able to evaluate the appropriateness of the REA model, perhaps identifying extensions. They may also be able to better articulate the characteristics of equity transactions which could improve other's understanding of their complexities, and which may identify new methods of designing, implementing and storing information about equity products. Thus, until the research is performed, it is unclear whether it will further our understanding of the REA model, the domain, or both.

Intangibles. Today's market realizes that a firm's production function is more complex than the simple exchanges we witness such as cash for inventory. Rather, a set of resources (such as goods, people with training, computer-enabled information, and advertising services) are exchanged for cash. For many firms the intangible assets in their production functions actually drive their financial success. For example, computer systems, brands, and human capital may be a firm's most valuable assets because they can provide differentiated goods and services. Yet today's financial reports fail to reflect the value of such assets by expensing costs associated with their creation, and the market has questioned the value of such traditional accounting measures (Lev 1997, Low, Siesfeld, and Larker 1999). As a result, accounting researchers are struggling to determine how to measure intangibles within organizations as is evidenced by the creation of the Vincent C. Ross Institute for Accounting Research and the Project for Research on Intangibles and the plethora of research being performed in this area.

We believe that REA provides insights into how accountants should recognize intangibles. For example, although current bookkeeping practices include the immediate expensing of service expenditures, such as advertising and consulting, since McCarthy (1982) the REA literature has advocated asset recognition and a separate event to reduce the quantity of the asset. For example, see the REA model of Advertising Service in Figure 2.

--- Insert Figure 2 approximately here ---

This model recognizes intangible assets because it follows the tenets of full-REA modeling discussed above (McCarthy 1982, David 1997, Geerts and McCarthy 2000a). First, the initial purchase of advertising service results in the recognition of an increase to the Advertising Service resource. Second, every resource must have at least one increment and decrement event related to it; therefore, the decrement event, Advertising Service Consumption, is needed. Third, every economic event must participate in at least one duality relationship. In Figure 2, management has determined that the advertising should result in a cash receipt from customers (as a result of a sale). Thus, this approach recognizes that the expenditure has been made to purchase something that adds value to the organization; in this case, the REA diagram explicitly shows that the cash receipt is the result of not only the sale of inventory, but also the effect of the advertising.

We believe this approach may be valuable for valuing other intangible assets such as human resource expenditures (Flamholtz 1974). For example, researchers should examine the possibility that employee training expenses be recognized as an asset, and then determine how that resource is deployed and used. Is it consumed through employee attrition or by the passage of time if the skills are transitory (as in technology related training)? Or do the skills gained actually appreciate over time? This approach could be evaluated for other intangible assets such as goodwill and brand recognition. In each case, the researcher must understand what expenditures are made to create the resource, and then make predictions of how it will be decremented, recognizing that the flows are not always explicitly visible or identifiable.

When these REA models are implemented, the resulting system would differ from a traditional system by enabling accountants to recognize the intangible value from the resources that have not traditionally been characterized as assets on the balance sheet. Additionally, explicit recognition of advanced production functions can help management with other decision-making situations. For example, to help to measure the value of intangible acquisitions, Hubbard (1997) recommends managers perform "thought experiments" in which they visualize the organization with and without the new purchase. If managers were familiar with using REA to model complex production functions, they would more rigorously identify all of the assets needed to implement the change and how those resources would add value.

ABC Costing. Several papers (such as Geerts 1993 and Grabski and Marsh 1994) have stated that there is a relationship between ABC costing and REA. In general, it has been hypothesized that ABC is a movement toward full recognition of expense drivers, and "full ABC" would be encompassed in full REA. Specifically, at one extreme, traditional accounting techniques have consolidated many expenses into an overhead account and then applied the overhead to products using one rate. REA, at the other extreme, requires that each expenditure result in an increase in another, explicit, asset (as was shown in figure 2). Methods to decrement each asset are identified, and the decrements are tracked individually. In the middle, ABC recognizes that multiple pools of assets are grouped together, and their cost drivers are used to decrement costs.

Although this relationship has been posited, only one study has provided rigorous research of the concept: Grabski and Marsh (1994) provide an overview of the relationship between these accounting techniques. However, we believe that a more detailed study describing explicitly how different decisions would result from an ABC system and an REA system would be valuable. We would hope to see one data set that was evaluated under both rule sets, and the specific differences between them explicated. This will require significant understanding of ABC costing techniques, and the identification of business environments in which the additional record keeping required by REA would be valuable. Note, however, that design scientists cannot be charged with determining the feasibility and cost effectiveness of the data collection techniques using today's technology. Rather, the researcher's objective should be to identify techniques that could be applied to improve the business performance and to provide guidance to the commercial community as to what features are necessary to implement such techniques. Commercial firms, with R&D resources, would then be charged with developing the technology to realize the new solutions. "The duty of the scholar is *to prepare the best possible models for various purposes and information needs, to put them at the disposal of the practitioners, and to educate the latter as to their relevance.* Whether practice will accept academic advice and use those models is a pragmatic and often political matter" (Mattessich 1995, 53).

Summary of REA extensions. We believe these three areas of research typify projects that would make valuable contributions to the AIS literature and meet the heuristics in McCarthy et al. (1992). The equity and intangible asset projects extensions reflect important problems to accountants that have been received very little attention in the REA literature. Therefore, as discussed earlier, these projects would likely constitute research, rather than design work. Because relatively little research has examined the relationship between ABC and REA systems, it is likely that there are opportunities for more research. However, we caution those attempting to extend the ABC studies to demonstrate their contributions by evaluating their new models against the prior ones. To be true contributions, these new implementations must make significant enhancements to our knowledge of REA and the domain. To successfully research complex components within these domains, rather than more trivial modules, researchers will need significant domain expertise. Therefore, it may be preferable to complete the projects described here in research pairs: one researcher with the accounting domain expertise and the other with knowledge of REA. These partnerships will help insure that resulting projects contribute to overall accounting knowledge and to the REA literature.

New Frontiers for REA Research

REA and Enterprise-Wide Information Systems. There are a number of different ways that design science research can contribute to our understanding of today's enterprise-wide information systems such as Enterprise Resource Planning (ERP) systems. As described in David, Dunn, McCarthy and Poston (1999), work is needed to determine the symbol sets used in today's commercially-available systems. This work may entail performing reverse engineering to determine what symbol sets were used to create the system, or alternatively, researchers could evaluate the match between an existing symbol set and a system. For example, because many universities are using ERP systems in their curricula, researchers could study these implementations to infer the underlying symbol sets or use system documentation to evaluate the completeness of the

REA model. O'Leary (1999c) has begun work in this area by comparing the basic REA patterns to SAP's ERP system. He found that the SAP implementation is, at a high level, consistent with the McCarthy (1982) conception of REA. However, we believe there is a need for more work at a more detailed level. If the researchers were to start from SAP and compare that with REA, their projects may highlight areas in which SAP is inconsistent with REA. Additionally, this approach may illuminate new features in SAP that have not been in present in previous REA evaluations.

Ideally, this work would be performed on several ERP systems. This would enable researchers to compare and contrast systems beyond their outwardly obvious characteristics such as their menu structure and the complexity of their user interfaces. Differences in business processes and database structure may, again, illuminate critical areas in data modeling that should be explored[4].

Research about ERP systems is necessary for several reasons. First, it can provide direct measures of the efficacy of the REA model. If researchers discover differences between the REA model and ERP systems, both will be evaluated. If the REA model is found to be missing key components, then either necessary extensions will be identified or its existing value will be rigorously evaluated. Additionally, if the systems do not conform to the REA template, recommendations for system enhancements can be made. However, to do this credibly, the researcher must be able to identify situations in which the current technologies provide poorer information than the

[4] Although there are hundreds of systems available, there are diminishing returns from evaluating multiple systems. Therefore, although the first and second evaluations are likely to be valuable, researchers must be very careful before examining yet another system. Rather, future evaluations should be driven by specific domain needs. These could include evaluating a new category of information system, or attempting to identify a specific system characteristic that has not been recognized, but that is needed to test a specific hypothesis.

recommended enhancements. Ideally, management decisions should be improved with the new system's information.

Second, if key differences are identified in the underlying symbol sets for more than one ERP system, researchers could evaluate different hypotheses implied by the design science REA literature. For example, David (1997) posits that ideally firms should not perform business or information events; rather they should develop processes so they could meet customer expectations through immediate economic events. The example that she uses is accounts payable systems, wherein REA would specify that payments be made based upon inventory receipts and that a vendor invoice is a redundant data source. If two ERP systems differ in how they process payables, with one processing vendor payments without requiring a vendor invoice and another requiring a vendor invoice, that would be a key difference between the systems. Research could explore whether firms benefited from using the more efficient processing, and what marketplace characteristics make the costs associated with processing vendor invoices necessary. The result of this body of literature would be a better understanding of REA (perhaps vendor invoices are needed), systems (how and when do they provide value), and organizational science (in what cases should each process be used).

ERP differences could then be explored using experimental or archival methods. For example, David and O'Donnell (forthcoming) identified several experimental research opportunities for studying ERP systems, but all required an understanding of the systems at their symbolic level. Additionally, once systems differences are identified, researchers can explore what company characteristics require the differential features, whether these features provide value, and, overall, how to successfully match firms with specific implementations.

Finally, the mapping between the theoretical model and implemented systems can add richness to our teaching environments. If REA and ERP systems are consistent, demonstrating this match in the classroom may incentivize students to struggle with learning difficult REA and systems concepts. More importantly, teaching REA will provide students with a conceptual framework that they can use when faced with new systems, so they will more quickly understand the complex environments they are likely to face. Future research should evaluate whether teaching REA and ERP are substitutes. If so, REA could be taught in place of ERP, saving universities the thousands of dollars necessary to implement such complex systems.

REA and Interorganizational Systems and Markets. It is obvious that the Internet is radically changing how business is conducted. What is less obvious is that Internet-based systems may be developed with fundamentally different architectures. Today organizations capture and store data about their transactions. Thus, researchers studying REA have focused on how to model the organization's operations to perform those activities most effectively. However, new classes of systems are rising in importance, and their models may be different from previous ones.

Organizations are gaining advantages by implementing advanced planning and supply chain execution systems such as i2 and Manugistics. To work most effectively these systems require data not only from the organization using them, but also from other trading partners in the supply chain. For example, to most effectively forecast demand, these systems are able to gather customer data. Similarly, to schedule productions, these systems can collect data from the company's suppliers to plan raw material acquisitions. Thus, these systems need *external* data models that span organizations, and, to date, very little work in REA has focused upon this extension. Therefore, we believe that this area of research needs to be explored in depth. We must determine if the exchange pattern that so effectively describes the environment when seen through one organization's eyes can be extended to recognize resources, events, and agents at other organizations (McCarthy 2000).

Perhaps more importantly, the growing importance of Application Solutions Providers (ASPs) and electronic market places could push the development of a radically different type of systems. Imagine that all firms in the supply chain conduct their business through one electronic marketplace such as being planned by the automobile industry (Hof, Welch, Arndt, Barrett, and Baker 2000). With today's technology, each firm in the supply chain would maintain its own copy of each transaction. Thus, a transaction between a supplier and a customer would be recorded twice: as a sale in the supplier's system, and as a purchase in the customer's. Now, imagine that there is one system for the marketplace and it stores all of the transactional data. To be most efficient, the transaction would only be stored once, yet each party in the transaction needs to be able to view the transaction from their own point of reference. For REA to work in this environment, many constructs may not apply exactly has they have in the past. For example, consider increment and decrement events -- if the transaction data are only being stored once, it is simultaneously an increment to one firm, and a decrement to another. How will this influence the system design? How do we identify which agent received the resource and which provided it? In addition, we believe the symbol set must be broadened to include constructs to "translate" the transaction so it appears to users in its appropriate form.

There are signs that this architecture may already be evolving in the marketplace. Agillion software provides Customer Relationship Management applications for small businesses (Kirkpatrick 1999). Rather than having the customers purchase and install the software, however, their product is completely Internet-based, and the data resides on Agilion's servers. Thus, when a customer *or* a supplier goes into the system, they are able to see all transactions that are related to their business. To date, the majority of this data is in the form of e-mail messages between the trading partners, but they have recently added a very simple sale transaction to the product features.

Additionally, although markets are being developed, there is some evidence that there are design weaknesses in them (Byrne, Lentz and Wolin 2000). They find that today's systems have been developed without a thorough understanding of the user's needs, and they believe that the market characteristics are changing. We believe that the REA framework may have even more relevance when electronic market places become "shared workspace sites" (393) that coordinate large projects between many types of vendors. These sites will require additional interorganizational capabilities such as bidding and project management facilities in addition to transaction processing. In that type of environment, we believe that REA may provide the open, semantic foundation that will be necessary (Haugen and McCarthy 2000).

Summary of Areas for REA Research. We believe the projects outlined are critical to extend the REA model and to furthering our understanding of today's systems. Ideally, we also believe that design science work in this area has the opportunity to influence tomorrow's systems. To make such a claim, however, we must proceed with extreme caution and rigor. To gain the necessary knowledge of today's systems and the theoretical knowledge of the REA model, it is likely necessary to borrow another practice from our computer science colleagues: we may need to work in teams to complete these projects. In computer science, universities often establish work groups that focus on an area of research. For example, MIT has been hugely successful with its Laboratory for Computer Science that it describes as follows:

> "The MIT Laboratory for Computer Science (LCS) is an interdepartmental laboratory whose principal goal is research in computer science and engineering. It is dedicated to the invention, development and understanding of information technologies which are expected to drive substantial technical and socio-economic change."
> http://www.lcs.mit.edu/about/

The scholars at such laboratories are able to collaborate and challenge each other's ideas to spark new innovations. By focusing on critical areas for development, they are able to build a comprehensive knowledge set, and thus they are able to make significant contributions to their literature and the economy. For example, the MIT LCS claims to have spawned over 35 firms that have influenced technological architectures. This is the type of success to which all design scientists should aspire.

Conclusions

A principal objective of this chapter is to provide an overview of the importance of design science research and to illustrate the concepts with one example from the AIS literature: the REA accounting model. Our goal was to help readers gain a better understanding of this research and its contributions. Whether researchers choose to perform design science research will be their decision. But when they read design science papers, we hope that they will now realize how design science and natural science can work together as Simon (1996) and March and Smith (1995) have so eloquently described. Additionally, reinforcing the McCarthy et al. (1992) guidelines for what constitutes research rather than development may help the readers evaluate the work of others.

Our second objective was to illustrate the importance of design science in the context of the REA literature. By consolidating most of the constructs, models, methods and instantiations in one place, we hope readers are able to synthesize the two decades of work here and to see the opportunities for extending it. The future research ideas that we have presented represent projects that we believe would be extremely interesting, rigorous, and rewarding. If, as an academic community, we were able to perform all of these projects, we believe the REA, information systems, and accounting literatures would all be enriched. However, this list of future projects is not exhaustive. In fact, as our knowledge base grows and technology advances continue, the possible future REA research projects are innumerable. Researchers are only cautioned to approach them with rich domain knowledge, a thorough understanding of REA, and an appreciation for the characteristics of high quality research. Armed with these attributes, their research efforts will likely prove successful.

As a final thought, we pose this challenge to AIS researchers: recognize that what has drawn us to design science research is the opportunity to shape the future of AIS. We, as design science researchers, can accept the responsibility for offering recommendations to practitioners. Although it is possible that practitioners will adopt only a fraction of the research recommendations made, there is a *chance* that our voices will change future information system designs. In fact, REA is beginning to reach a wider audience than AIS academics. Major ERP vendors have called to discuss our REA work, IBM has implemented a system following the REA pattern (Cherrington, Denna, and Andros 1996 and Andros, Cherrington, and Denna 1992), and a supply chain system is developing based exclusively upon REA (Haugen and McCarthy 2000). These developments demonstrate that the ideas embodied in this research have merit, and we believe that future, high-quality research projects will contribute to both the literature and the economy!

REFERENCES

Adamson, I. L. and D. M. Dilts. 1995. Development of an accounting object model from accounting transactions. *Journal of Information Systems* (Spring): 1-25.

Andros, D.P., J.O. Cherrington, and E.L. Denna. 1992. Reengineer your accounting, the IBM way. *Financial Executive* (July/August): 28-31.

Anonymous. 2000. MIT Laboratory for Computer Science. URL **http://www.lcs.mit.edu/about/**.

Armitage, H. M. 1985. *Linking management accounting systems with computer technology*. Hamilton, Ontario: Society of Management Accountants of Canada

Batini, C., S. Ceri, and S.B. Navathe. 1992. *Conceptual Database Design: An Entity Relationship Approach*. Redwood City, California: The Benjamin/Cummings Publishing Company, Inc.

Bubenko, J. 1977. The temporal dimension in information modeling. In G. M. Nijssen (ed.) *Architecture and Models in Data Base Management Systems*. North Holland. Amsterdam. 93-118.

Byrne, T., N. Lentz, and S. Wolin. 2000. Beyond the exchange. *Business 2.0.* (6:13) 390-393.

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) 560-75.

Chen, N. and D. McLeod (1995). Schema evolution for object-based accounting database systems. *Expert Systems with Applications* (volume 4, no. 9) 491-502.

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

Cherrington, J.O., E.L. Denna, and D.P. Andros. 1996. Developing an event-based business solution: The case of IBM's national employee disbursement system. *Journal of Information Systems* (Spring): 51-69.

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

Colantoni, C.S., R.P. Manes, and A. Whinston. 1971. A unified approach to the theory of accounting and information systems. *The Accounting Review* (January): 90-102.

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.

David, J.S., C.L. Dunn, W.E. McCarthy, and R.S. Poston. 1999. The research pyramid: A framework for accounting information systems research. *Journal of Information Systems* (13:1 Spring): 7-30.

David, J.S. and E. O'Donnell. Forthcoming. How Information Systems Influence User Decisions: A Research Framework and Literature Review. *International Journal of Accounting Information Systems*.

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

Denna, E.L., J. Jasperson, K. Fong, and D. Middleman. 1994. Modeling conversion process events. *Journal of Information Systems* (Spring): 43-54.

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

Dunn, C.L. 1994. An Investigation of Abstraction in Events-based Accounting Systems. Doctoral Dissertation. Michigan State University.

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.

Earman, J. 1992. *Bayes or Bust*. Cambridge, Massachusetts: MIT Press.

Elmasri, R. and S.B. Navathe. 1994. *Fundamentals of Database Systems*, Second Edition. Menlo Park, CA: Addison-Wesley Publishing Company.

Ernst, G.W. and A. Newell. 1969. *GPS: A Case Study in Generality and Problem Solving*. New York, NY: Academic Press.

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

Flamholtz, E. G. 1974. *Human Resource Accounting*. Dickenson Publishing Company. Encino, CA.

Gal, G. and W.E. McCarthy. 1982. Operation of a Relational Accounting System. Presentation to the Northeast American Accounting Association Meeting, Newport, R.I., April 1982.

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: North-Holland.

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

Gal, G. and W.E. McCarthy. 1986. Operation of a relational accounting system. *Advances in Accounting* 3: 83-112.

Gal, G. and W.E. McCarthy. 1991 "Semantic Specification and Automated Enforcement of Internal Control Procedures Within Accounting Systems," at the European Accounting Congress, Maastricht, The Netherlands, April.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1995. *Design Patterns : Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

Geerts, G. 1993. Toward a new paradigm in structuring and processing accounting data. Doctoral Dissertation. Free University of Brussels.

Geerts, G. and W.E. McCarthy. 1992. The extended use of intensional reasoning and epistemologically adequate representations in knowledge-based accounting systems. *Proceedings of the Twelfth International Workshop on Expert Systems and Their Applications,* Avignon, France (June): 321-32.

Geerts, G. and W.E. McCarthy. 1994. The economic and strategic structure of REA accounting systems. 300th Anniversary Program, Martin Luther King University, Halle-Wittenberg, Germany.

Geerts, G. and W.E. McCarthy. 1997a. 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. Springer-Verlag.

Geerts, G. and W.E. McCarthy. 1997b. Using object templates from the REA accounting model to engineer business processes and tasks. Paper presented to the 20th Congress of the European Accounting Association, Graz, Austria, April.

Geerts, G. and W.E. McCarthy. 1997c. Accounting as Romance: Patterns of Unrequited Love and Incomplete Exchanges in Life and in Business Software. Presentation to the *Workshop on Business Object Design and Implementation- II* (October 1997) at the ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA 97), Atlanta, Georgia.

Geerts, G. and W.E. McCarthy. 1997d. Use of an Accounting Object Infrastructure for Knowledge-Based Enterprise Models. Presentation to the *AAAI Spring Symposium on Ontological Engineering*, Stanford University (March).

Geerts, G. and W.E. McCarthy. 1999. An Accounting Object Infrastructure For Knowledge-Based Enterprise Models," *IEEE Intelligent Systems & Their Applications* (July/August 1999), pp. 89-94

Geerts, G. and W.E. McCarthy. 2000a. "Augmented Intensional Reasoning in Knowledge-Based Accounting Systems," forthcoming in *The Journal of Information Systems*

Geerts, G. and W.E. McCarthy. 2000b. An Ontological Analysis of the Primitives of the Extended-REA Enterprise Information Architecture," forthcoming in *The International Journal of Accounting Information Systems*

Geerts, G. and W.E. McCarthy. 2000c. The Ontological Foundation of REA Enterprise Information Systems, Presentation to The Annual Meeting of The American Accounting Association, Philadelphia, August.

Geerts, G., W.E. McCarthy, and S. Rockwell. 1996. Automated integration of enterprise accounting models throughout the systems development life cycle. *International Journal of Intelligent Systems in Accounting, Management and Finance* (September): 113-128.

Guarino,N. 1998. Formal ontology and information systems. *Proceedings of FOIS'98*, Trento, Italy.

Grabski, S.V. and R.J. Marsh. 1994. Integrating accounting and manufacturing information systems: An ABC and REA-based approach. *Journal of Information Systems* (Fall): 61-80.

Haseman, W.D. and A.B. Whinston. 1977. *Introduction to Data Management*. Homewood, IL: Richard D. Irwin.

Haugen, R. 1997. Radically distributed supply chain systems. OOPSLA'97 Workshop on Business Object Design and Implementation III, Atlanta, GA.

Haugen, R. and W.E. McCarthy. 2000. REA, a semantic model for Internet supply chain collaboration. URL **www.supplychainlinks.com/Rea4scm.htm**.

Hines, R.D. 1988. Popper's methodology of falsificationism and accounting research. *The Accounting Review* (October): 657-662.

Hof, R.D., D. Welch, M. Arndt, A. Barrett, and S. Baker. 2000. E-malls for business. *BusinessWeek* (March 13) URL **http://www.businessweek.com/2000/00_11/b3672063.htm**

Hubbard, D. 1997. Everything is measurable. *CIO Magazine* (November 15) URL **http://www.cio.com/archive/enterprise/111597_checks_content.html**

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

Kirkpatrick, D. 1999. Enterprise software so simple even you can use it! Dumb name, cool company. *Fortune.* (140: 8) 344.

Lastrucci, C.L. 1963. *The Scientific Approach: Basic Principles of the Scientific Method*. Cambridge, Massachusetts: Schenkman Publishing Company, Inc.

Lev, B. 1997. The old rules no longer apply. *Forbes ASAP* (April 7) URL **http://www.forbes.com/asap/97/0407/034.htm**

Low, J., T. Siesfeld, and D. Larker. 1999. Capital Thinking. *Forbes ASAP* (August 3) URL **http://www.forbes.com/asap/99/0823/141.htm**

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) 328-39.

March, S.T. and G.F. Smith. 1995. Design and natural science research on information technology. *Decision Support Systems* 15: 251-266.

Mattessich, R. 1995. *Critique of Accounting: Examination of the Foundations and Normative Structure of an Applied Discipline*. Westport, Connecticut: Quorum Books.

McCarthy, J. and P.J. Hayes. 1969. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie (Eds.), *Machine Intelligence 4*, New York, NY: American Elsevier, 463-502.

McCarthy, W.E. 1979. An entity-relationship view of accounting models. *The Accounting Review* (October): 667-686.

McCarthy, W.E. 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) 625-37.

McCarthy, W.E. 1982. The REA accounting model: A generalized framework for accounting systems in a shared data environment. *The Accounting Review* (July): 554-578.

McCarthy, W.E. 1984. Materialization of account balances in the REA accounting model. Annual Meeting of the British Accounting Association, Norwich, England.

McCarthy, W.E. 1997**.** An ontological analysis of the REA accounting domain model. Presentation made to the *Conference on Ontological Perspectives in Information Systems*, Balzano, Italy, September 1997

McCarthy, W.E. 2000. REA Accounting models -- the need for both internal and external views in a world of interconnected enterprises. Keynote presentation to the Third European Conference on Accounting Information Systems. Munich (March).

McCarthy, W.E., E. Denna, G. Gal, and S.R. Rockwell. 1992. Expert systems and AI-based decision support in auditing: progress and perspectives. *Intelligent Systems in Accounting, Finance and Management* 1: 53-63.

McCarthy, W.E. and S. Rockwell. 1989. The integrated use of first-order theories, reconstructive expertise, and implementation heuristics in an accounting information system design tool. *Proceedings of the Ninth International Workshop on Expert Systems and Their Applications*. Avignon, France, EC2: 537-548.

McCarthy, W.E., S. Rockwell, and H.M. Armitage. 1989. A structured methodology for the design of accounting transaction systems in a shared data environment. *Proceedings of the 1989 Conference of the Structured Techniques Association* May: 194-207. Chicago, Illinois.

McGann, J. and D. Seaman. n.d./2000. British Poetry 1780-1910: a Hypertext Archive of Scholarly Editions. URL http://etext.lib.virginia.edu/britpo.html

McJones, P. n.d./2000. System R. URL http://www.mcjones.org/System_R/

Nakamura, H. and R.E. Johnson. 1998. "Adaptive Framework for the REA Accounting Model," in *Proceedings of OOPSLA'98 Business Object Workshop IV,* J. Sutherland (ed.), 1998, Vancouver.

Newell, A. and H.A. Simon. 1976. Computer science as empirical inquiry: symbols and search. *Communications of the ACM* 19: 113-126.

O'Leary, D.E. 1988. Software engineering and research issues in accounting information systems. *Journal of Information Systems* (Spring): 24-38.

O'Leary, D.E. 1999a. REAL-D: A schema for data warehouses. *Journal of Information Systems* (Spring): 49-62.

O'Leary. D. 1999b. Modeling time in REA/REAL databases. Working paper. University of Southern California.

O'Leary D., 1999c. On the Relationship Between REA and SAP. Working paper. University of Southern California.

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

Rockwell, S. R. 1992. The conceptual modeling and automated use of reconstructive accounting domain knowledge. Unpublished doctoral dissertation, Michigan State University.

Rockwell, S. R. and W. E. McCarthy. 1999. REACH: Automated Database Design Integrating First-Order Theories, Reconstructive Expertise, and Implementation Heuristics for Accounting Information Systems" in *International Journal of Intelligent Systems in Accounting, Management, and Finance,* September.

Sorter, G.H. 1969. An 'events' approach to basic accounting theory. *The Accounting Review* (January): 12-19.

Seddon, P.B. 1996. An architecture for future computer-based accounting systems: Generating formula accounting journal entries from TPS databases using the resources and exchange events accounting model. *Journal of Information Systems* (Spring): 1-25.

Simon, H.A. 1969. *The Sciences of the Artificial*. Cambridge, Massachusetts: The M.I.T. Press.

Simon, H.A. 1995. Artificial intelligence: an empirical science. *Artificial Intelligence* 77: 95-127.

Simon, H.A. 1996. *The Sciences of the Artificial*, Third Edition. Cambridge, Massachusetts: The MIT Press.

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

Sowa, J.F. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA.: Addison-Wesley.

Sowa, J. 1999. *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* Brooks/Cole Publishing, Pacific Grove, CA.

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

Verdaasdonk P. J. A. 1999. Accounting information for operations management decisions. Ph.D. thesis, Eindhoven University of Technology.

Verdaasdonk P. J. A. 2000. An object model for ex ante accounting data, working paper, Eindhoven University of Technology.

Walker, K.B. and E.L. Denna. 1997. Arrivederci, Pacioli? A new accounting system is emerging. *Management Accounting* 1 (July): 22-30.

Zloof, M. M. 1975. Query-by-Example. *Proceedings of the National Computer Conference* (AFIPS 1975) 431-37.

Figure 1

REA Model of Equity Transactions

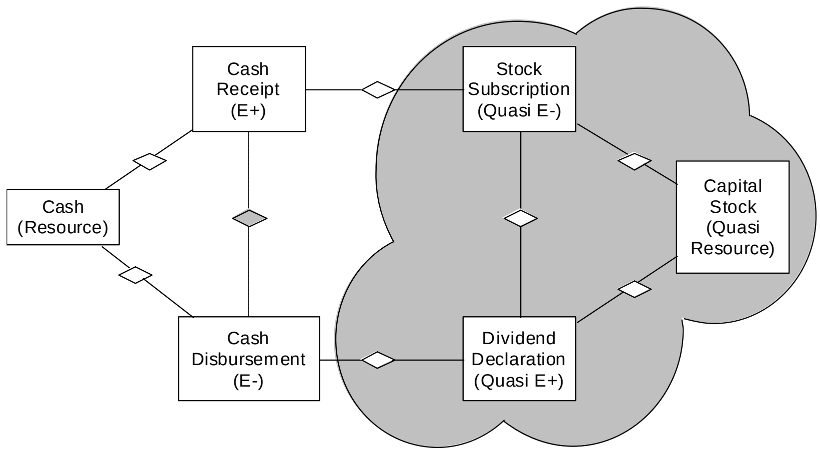


Figure 1: REA Model of Equity Transactions

Adapted from McCarthy (1982, 1984)

Figure 2

REA Model of Resource Recognition

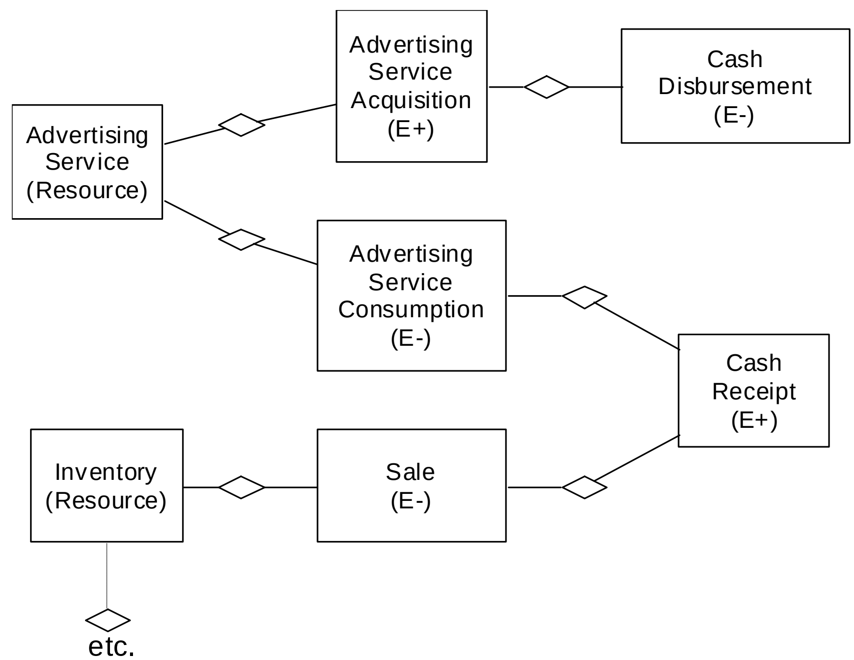


Figure 2: REA Model of Resource Recognition.

Adapted from McCarthy (1982)

Table 1

Generalization of Research sub-groups in AIS

Social Scientist	Computer Scientist
Descriptive scholarly activities	Prescriptive scholarly activities
Natural science	Design science
Positive philosophy	Normative philosophy
Discover	Create

Table 2
March and Smith's (1995, 255) Research Framework

Research Activities

Research Outputs

	Build	Evaluate	Theorize	Justify
Constructs				
Model				
Method				
Instantiation				

Table 3 Design Science Papers or Books That Have Influenced REA Research

PAPER	TOPIC	IMPLICATION
Codd (1970)	relational database model, relational languages, normalization	seminal paper on logical design of database structures and theoretical use of database procedures
Colantoni, Manes, and Whinston (1971)	database accounting	first paper to outline architecture for database-oriented accounting systems
Chamberlin et al. (1976)	procedural aspects of databases, SEQUEL language	overview of operators and procedures for the most widely used database language (SQL)
Chen (1976)	Entity-Relationship model, database semantics	seminal paper on semantic database design
Everest and Weber (1977)	relational design of traditional accounting constructs	identified overt difficulties with representation artifacts of double-entry accounting
Smith and Smith (1977)	generalization hierarchies, typification abstractions	introduced the idea of generalization to databases and pioneered the integrated use of aggregation and generalization abstractions in design
Bubenko (1977)	conclusion materialization; temporal database dimensions	explored the general difficulties involved in adapting the concepts of time to structured databases
Lum et al. (1979)	New Orleans database design methodology	seminal paper for the most widely-accepted phase definitions of database design (requirements analysis, conceptual design, logical design, physical design)
Tsichritzis and Lochovsky (1982)	declarative-procedural-constraint categorization, navigational and specificational procedure definition	definitive text on the categorization of features of semantic and syntactic database design and use
Sowa (1984)	declarative-procedural representation, conceptual relativity, knowledge-based design	definitive text on the philosophical, psychological, and linguistic foundations of conceptual modeling
McCarthy and Hayes (1969)	epistemological adequacy, intensional reasoning	defined the metrics for different classes of knowledge-based systems
Porter (1985)	value chains, business process differentiation	seminal text on the use value chains and value systems in strategic planning
Gamma et al. (1995)	object-orientation, design patterns	seminal text on the use of design patterns in object-oriented development of information systems
Sowa (1999)	ontology constructs and agent use	definitive text on the conceptual foundations of ontologies, agents, logic programming, and knowledge representation

Table 4 -- Papers That Made Significant Design Science Advances In REA Modeling

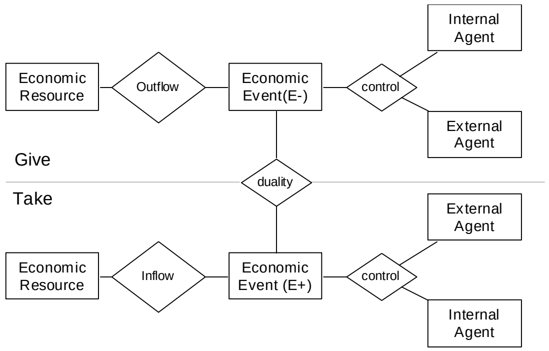
PAPER	TOPIC	IMPLICATION
McCarthy (1979;1980)	Entity-relationship modeling	Use of explicit semantics in designing accounting systems and relational language specification (SEQUEL) of accounting operations
McCarthy (1982)	Resource-Event-Agent model	Created a transaction pattern for economic exchanges without classificational double-entry artifacts and reconciled the persistent use of that pattern to conventional accounting procedures
Gal and McCarthy (1983)	Network database implementation (CODASYL)	Working prototype of a database-oriented accounting system with navigational procedures
Gal and McCarthy (1986)	Relational database implementation (Query-by-Example)	Working prototype with relational (specification) language, with exploration of set-oriented difficulties with accounting data, and with hierarchical materialization of account balances
Denna and McCarthy (1987)	Decision support systems	Personalization of REA database to decision needs of particular managers with views, graphics, spreadsheets
Rockwell and McCarthy (1989;1999)	CASE tool for accounting database design	Use of domain-specific accounting knowledge for view modeling, view integration, and implementation compromise
Geerts and McCarthy (1992; 2000a)	Intensional reasoning and epistemological adequacy defined in the context of CREASY system	Definition of full-REA models and extension of semantic frameworks from design (passive schemas) to operation (active schemas)
Geerts and McCarthy (1994;1997a;1999)	Abstraction of exchange patterns to business processes and enterprise value chains	Formal models of enterprise-wide tracking of economic transaction data and of business processes as production functions with patterned representation
Dunn and McCarthy (1997)	Definition of database orientation, semantic orientation, and structuring orientation	Established criteria for differentiating different classes of accounting systems
Geerts and McCarthy (2000b, 2000c)	Ontological extensions to REA models with types and commitments	Expanded the definitions of REA primitives to include additional entities (types, commitments, exchange) and relationships (association, custody, reserves, executes, reciprocal)

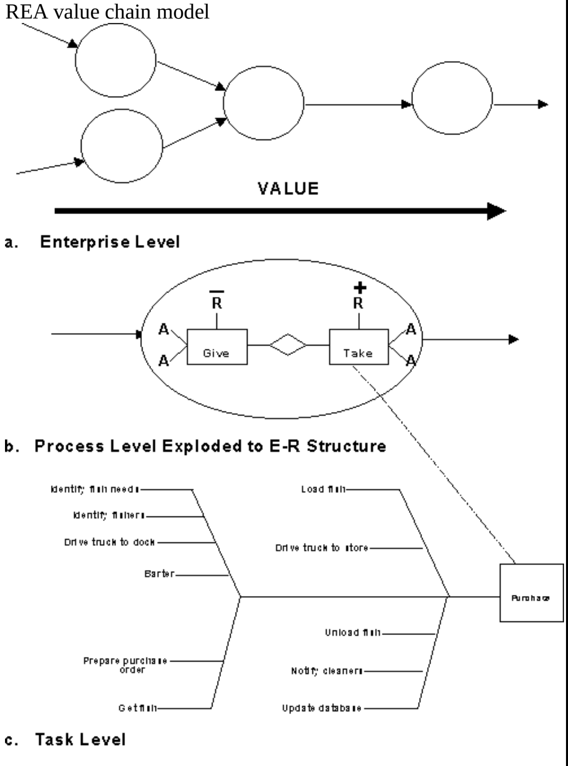
Table 3: Significant Constructs, Models, Methods, and Instantiations Derived from REA Design Science Research

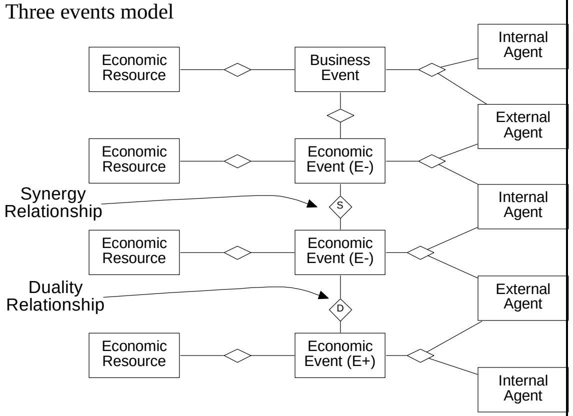
Constructs		
PAPERS	**DESIGN SCIENCE RESEARCH OUTPUTS**	**DEFINITION or DESCRIPTION (as adapted from cited source)**
McCarthy (1982)	Economic resources	Economic resources are defined by Ijiri [1975, pp. 51-2] to be objects that (1) are scarce and have utility and (2) are under the control of an enterprise. 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 (McCarthy 1982, 562).
	Economic events	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." McCarthy (1982, 562). McCarthy (1982,562) additionally suggests that theoretically "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."
	Economic agents	Economic agents include persons and agencies who participate in the economic events of the enterprise or who are responsible 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 (McCarthy 1982, 562).
	Economic units	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 (McCarthy 1982, 563).
	Stock-flow relationships	Stock-flow relationships simply connect the appropriate elements in the entity sets defined above [i.e., the economic resources and events]. 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 (McCarthy 1982, 562).
	Duality relationships	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). (McCarthy 1982, 562).
	Control relationships	Control relationships are 3-way associations among (1) a resource increment/decrement (event), (2) an inside party (unit), and (3) an outside party (agent). 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" (McCarthy 1982, 564). Readers should note that control relationships may be modeled using either one ternary or two binary relationships.
	Responsibility relationships	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 (McCarthy 1982, 565).
	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 (McCarthy 1982, 567). Conclusion materialization is a concept adapted from Bubenko (1977) and it is used, in part, to generate account balances procedurally if needed.
	Claims	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) McCarthy (1982, 568).
	Declarations -- Procedures	The declarative features of an accounting schema consist of its base objects -- those elements representing economic events, resources, and agents plus relationships between them (McCarthy 1982, 569). Also see the construct procedures. The procedural features consist of methods for materializing conclusions about base objects (McCarthy 1982, 569).
Gal and McCarthy (1984, 1991)	Internal control constraints	Database constraints configure how one representation is allowed to transition into another according a given set of transformation or business rules. For accounting systems, that rules often equate to internal control specifications.
Denna et al.. (1993)	Locations	Denna et al. (1993, 60-1) assert that "to the extent that it is important, we should make sure data about the location of an event are captured. Sometimes the location of the event is embedded in the location of the agents or resources involved. However, when the event location is not derivable by association with the resources or agents, we must explicitly specify the event location."

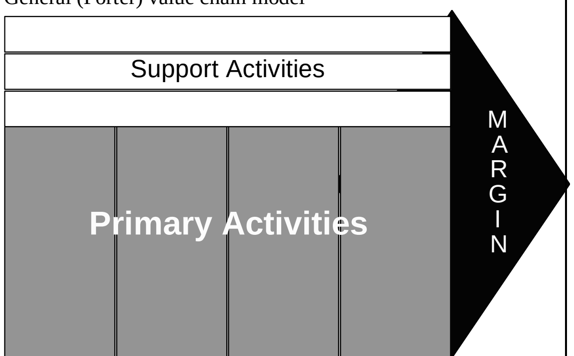
Constructs		
PAPERS	**DESIGN SCIENCE RESEARCH OUTPUTS**	**DEFINITION or DESCRIPTION (as adapted from cited source)**
Rockwell and McCarthy (1999)	Implementation compromise	Implementation compromise refers to the trade-offs that occur when implementing an REA-based accounting system that does not meet the definition of Full-REA accounting. According to Rockwell and McCarthy (1999, 189) there are two categories of trade-offs: "(a) compromises based upon information use characteristics, and (b) compromises based upon physical implementation characteristics." Use of both of these categories is strongly predicated on REA pattern matches.
Geerts and McCarthy (1997)	Process	A process encompasses two mirror-image REA patterns, one an increment and the other a decrement, connected by a duality relationship. At the process level, the decremented resource is the input while the incremented resource is the output. A process thus defined is equivalent to an economic production function.
	Value chain	A value chain is a purposeful sequence of business processes where the factors of production are acquired, transformed into value-added products or services, and then delivered to customers. The interplay between the original REA primitives and the Porter notion of a value chain is best explained in Geerts and McCarthy (1997, 98). "Taken as a whole, duality relationships are the glue that binds a firm's separate economic events together into rational economic processes, while stock-flow relationships weave these processes together into an enterprise value chain (Porter 1985; Geerts and McCarthy, 1994) or scenario (Geerts (1993). In its most general form, a value chain … is a purposeful set of economic exchanges where an initial outlay of cash is successively converted into some types of more valuable intermediate resource and then finally converted back to cash."
	Task	"Tasks in REA analysis are, by definition, compromises to full specification (that is, they are economic events where an analyst doesn't try to specify full patterns)" (Geerts and McCarthy, 1997, 98)
David (2000)	Business event	Business events are defined as "any business activity that management wants to plan, monitor, and evaluate" (Denna et al. 47). These events result in changes to the physical world and provide new information that can be used by the firm's management to make decisions (David 2000, 12).
	Information event	Information events are defined as "procedures that are performed in organizations solely to capture, manipulate, or communicate information." The key distinction between these events and business and economic events is that no new data is identified (although the previously identified data may be captured or summarized and reported), and nothing changes in the physical that the REA diagram has not already described. This type of event includes the specific implementation methods for capturing data about the resources, events, and agents, as well as any report generation performed with the data in the system (David 2000, 15).
	Synergy relationship	Synergy relationships link multiple events of a similar nature, usually decrements. This enables modelers to represent "bundles" of activities that are performed to meet an objective
Dunn and McCarthy (1997)	Database orientation	A database orientation as defined here requires three conditions: 1. Data must be stored at their most primitive levels (at least for some period), 2. Data must be stored such that all authorized decision makers have access to it, and 3. Data must be stored such that it may be retrieved in various formats as needed for different purposes. These conditions do not require the use of database technology – object oriented, artificial intelligence, or other technologies that allow storage and maintenance of primitive detail accommodate this orientation. This also allows for systems built using database technology that do not have a database orientation (Dunn and McCarthy 1997, 36).
	Semantic orientation	Integrated semantics is a fundamental idea of modern database management, reflected in Abrial's (1974, 3) definition, "a database is a model of an evolving physical reality." Re-stated in terms of design methodology, this means that all potential users of a database pool their notions of important information concepts and use that integrated set of ideas to build one conceptual data model that serves everybody. The objects in this conceptual model are required to correspond closely to real world phenomena, hence the accentuated use of the term semantic to describe this activity. In an accounting domain, integrated semantics means that accounting models should depict the economic exchanges or processes that produce the firm's accounting data…components of the models should reflect real world phenomena, a situation that precludes the use of basic double-entry artifacts (e.g., debits, credits, accounts) as declarative primitives. Semantically modeled accounting systems allow representations of economic exchange phenomena to be integrated well with descriptions of non-accounting phenomena (Dunn and McCarthy 1997, 37).
	Structuring orientation	A structuring orientation mandates the repeated use of an occurrence template as a foundation or accountability infrastructure for the integrated business information system. There are two core structuring ideas within the REA accounting model. First is the use of a template that records and stores data associated with sets of economic events…for each economic event, data are recorded and stored pertaining to resources and agent connected to the event…The REA model also requires that data about relationships between or among the entities be maintained. Therefore, the data must be stored in such a way that the links (1) between an event and its resources involving inflows and outflows (stock-flow relationships) and (2) among an event and its agents involving participation (control relationships) are preserved. The second structuring idea is that there are two basic types of economic events – resource outflows (give) and resource inflows (take) – and that these types are normally coupled through duality relationships. The structuring orientation of REA accounting enables the maintenance of a centralized data bank, structured such that the resulting accounting system can serve as a framework for the

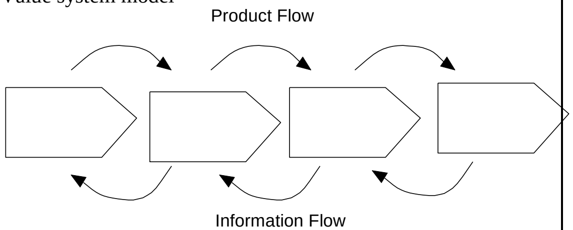
Constructs		
PAPERS	**DESIGN SCIENCE RESEARCH OUTPUTS**	**DEFINITION or DESCRIPTION (as adapted from cited source)**
		integrated business information system (Dunn and McCarthy 1997, 37).
Geerts and McCarthy (2000a)	Epistemological adequacy	The idea of epistemological adequacy – first described by McCarthy and Hayes (1969) – is a notion familiar to most AI theorists, but the application of this concept to accounting by Geerts and McCarthy is somewhat unique. This is their heuristic for determining such adequacy: "if a representation allows the full extent of intensional reasoning in materializing data-dependent conclusions and in enforcing integrity constraints, we consider epistemological features adequate. Anything less means that we strive for a higher degree of representational faithfulness" (Geerts and McCarthy 2000a). For REA, epistemological adequacy is provided when all of the entities and relationships of the basic pattern are instantiated throughout the database; this condition is called full-REA.
	Intensional reasoning	Intensional reasoning is pattern-matching logic that assumes full-REA representation where all components of the basic REA pattern are represented declaratively in full. The procedural logic is intended to work at the type level (the database intension) as opposed to the normal case where logic works at the token level (the database extension). The definition of claim in Geerts and McCarthy (2000) is a good example of intensional reasoning.
	Full-REA modeling	Full-REA modeling is what occurs when the metric of epistmological adequacy is applied to REA. Its definition is dependent upon full compliance with the basic entities and relationships of the model: (1) all increment events need to be linked to decrement events and vice-versa, (2) all resources must be materialized and both their inflow and outflow events identified, and (3) all events need at least one inside and outside agent as they are assumed to be part of an exchange that occurs at arm's length between parties with competing economic interests.
Geerts and McCarthy (2000b)	Type image	A type image is the abstract characterization of any of the basic REA entities via grouping (Sowa 1999). For example, inventory may be grouped into product families; orders may be grouped into immediate fills, backorders, and returned orders; and customers may be grouped into market segments.
	Commitment image	A commitment image is a precursor to an economic event where a party agrees to engage in a resource transfer at within a scheduled time in the future. Examples of commitments are sale orders, hotel and airplane reservations, production orders, and raw material requisitions.
	Accountability infrastructure & policy infrastructure	The accountability infrastructure of a firm conceptualizes its full history of obtaining initial financing, of using that financing to acquire and deploy the factors of production, and finally of using the results of that production to satisfy customers and to become profitable The policy infrastructure on the other hand conceptualizes what "could be" or "should be" within the context of a defined portfolio of firm resources and capabilities.

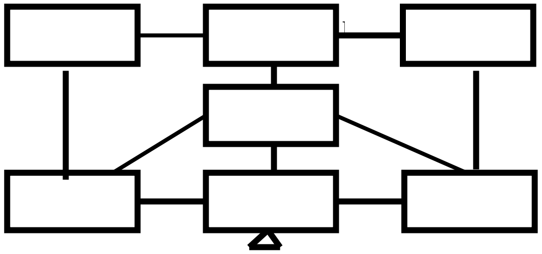
Models		
PAPERS	**DESIGN SCIENCE RESEARCH OUTPUTS**	**DEFINITION or DESCRIPTION (as adapted from cited source)**
McCarthy (1982)	REA model 	The basic REA model that shows each economic event, the resource(s) that are being incremented and decremented, and the agents who are participating. The duality relationship represents the exchange, i.e. it shows what resource was increased in exchange for decreasing another. The model also differentiates between internal and external agents for an event, and it establishes recursive relationships among economic units (inside agents). The three-way control relationship is routinely broken into two binary relationships for simplicity purposes.

Models		
PAPERS	**DESIGN SCIENCE RESEARCH OUTPUTS**	**DEFINITION or DESCRIPTION (as adapted from cited source)**
Geerts and McCarthy (1997a)	REA value chain model  Figure 2 -- Different REA Abstraction Levels	This model includes three levels of abstraction of an organization's operations. At the highest level, processes are represented as simple bubbles, with arrows into the bubble representing what is consumed in the exchange, and arrows out for what is created. Examples include the revenue and procurement processes. This level is similar to Porter's value chain in that the focus is showing high level processes that add value. The difference is the explicit recognition of the resources that are consumed and produced. The middle layer of abstraction is the basic REA model that was discussed above. The lowest level of abstraction shows a "fishbone" diagram of the tasks performed to complete the business process. At this level of detail, the REA template is no longer enforced because, at this level, it is very difficult to trace all of the resources associated with each activity. Rather, managers may choose to combine all of the costs associated with these activities and apply them to the overall process. Examples of tasks would include Taking a Customer Order, Performing Credit Check, and Generate Sales Reports.

<table>
<tr><th colspan="3">Models</th></tr>
<tr><th>PAPERS</th><th>DESIGN SCIENCE RESEARCH OUTPUTS</th><th>DEFINITION or DESCRIPTION (as adapted from cited source)</th></tr>
<tr><td>David (2000)</td><td>Three events model
</td><td>This model is an extension of the basic REA pattern. Business events represent a subset of the tasks represented in the Geerts and McCarthy (1997a) model. Like tasks, they do not participate in duality relationships. However, because these events would be implemented in any resulting information systems, they are used only to represent events that add new information that is valuable to management. However, they are not classified as economic events because they do not add value in the Porter (1985) sense of the word. Commitments are examples of business events. Information events would be included as tasks in the Geerts and McCarthy (1997a) model, but would be omitted from a three events model.

This model also introduces synergy relationships to link multiple events of a similar nature, usually decrement events. This enables modelers to represent "bundles" of activities that are performed to meet an objective. For example, if one provides a customer with goods and a 30 day service contract, there could be two decrementing economic events: Sale and Provide Service, and the Cash Receipt is really for both of these. Therefore, the two should be related, and it is assumed that the customers will value then together as greater than the sum of them separately.</td></tr>
<tr><td>Porter (1985)</td><td>General (Porter) value chain model
</td><td>The primary activities in this model represent how firms create value for their customers. Such activities include inbound and outbound logistics, manufacturing, sales and service. Support activities do not directly add value to customers and include accounting, human resources, and information technology.

This model has been used to help support activities focus their attention on providing value to those in primary functions in order to create value for the firm. They have also been used to guide integration efforts both through process reengineering and implementation of information technology. In this role, they are used to flush out activities that do not add value and to identify links between activities that can be automated to improve the information flow.</td></tr>
<tr><td></td><td>Value system model
</td><td>These models represent a firm's supply chain. They recognize that every company from a raw material supplier through the final customer is responsible for its own value chain (as each company is shaped as a value chain), but that relationships between the firms need to be evaluated to improve efficiency.</td></tr>
</table>

Models		
PAPERS	**DESIGN SCIENCE RESEARCH OUTPUTS**	**DEFINITION or DESCRIPTION (as adapted from cited source)**
Geerts and McCarthy (2000b)	The REA Ontology 	The lowest level boxes represent the economic resources, economic events, and economic agents from the basic model. These are augmented first by *commitment images* for events (middle box), and then by *type-images* for all entities. The three boxes at the lowest level form the accountability infrastructure while those at the second and third level constitute the policy infrastructure. Additionally, the ontology describes multiple new instances of needed relationships.

Methods		
PAPERS	**DESIGN SCIENCE RESEARCH OUTPUTS**	**DEFINITION or DESCRIPTION (as adapted from cited source)**
McCarthy (1979)	Semantic modeling of accounting phenomena	Semantic modeling (as exemplified by the E-R methodology of Chen (1976)) is a process that constructs the declarative features of a database by abstracting directly from elements of the object system to be modeled (semantic = reality to symbol mapping). Its syntactic complement is normalization which optimizes declarations by restructuring based upon the set of functional dependencies among existing representations (syntax = symbol to symbol mapping).
McCarthy (1982)	Pattern-driven view modeling and view integration	Augmented the New Orleans database design methodology with the notion of using a specific framework -- the REA model -- to guide both view modeling (a template for attributes) and view integration (identifying integration points).
Gal and McCarthy (1983)	Network database methods (CODASYL)	Developed methods for building REA models with network databases, including adapting m-n relationships to owner-coupled sets and adapting the need for triggered updates to navigational programming
Gal and McCarthy (1986)	Relational database methods	Developed methods for building REA models with specificational procedures, including adapting the use of special algebra operations like set difference operators to materializing accounting conclusions. Identified problems associated with specificational (set oriented) retrieval (no use of LIFO or FIFO possible) and problems associated with null values in materializing claims.
Gal and McCarthy (1986), McCarthy (1984)	Conclusion materialization methods	Developed the notion of materializing a chart of accounts hierarchically in accordance with definitions derived from both the FASB and the original REA paper. Account balances were thus moved entirely from declarative representations into procedural ones.
Rockwell and McCarthy (1999)	Implementation compromise methods	Developed pattern-matched cost-benefit heuristics for compromising REA models with results that map into commercially-available implementations
Nakamura and Johnson (1998)	Design patterns	Identified how the materialization of accounting data in an REA-oriented system could be expedited with the use of specific object-oriented design patterns
O'Leary (1999a)	Data warehouse adaptations	Identified methods for adapting REA modeling to the task of building data warehouses

Instantiations		
PAPERS	**DESIGN SCIENCE RESEARCH OUTPUTS**	**DEFINITION or DESCRIPTION (as adapted from cited source)**
Armitage (1985)	QBE implementation	REA-oriented system dealing with manufacturing was implemented in QBE to illustrate the efficacy of REA for producing managerial decision data
Rockwell and McCarthy (1999)	REACH (computer-aided software engineering tool)	REACH is designed to aid in the process of database design in general and in the sub-processes of view modeling and view integration in particular. To do this, REACH uses three kinds of accounting domain knowledge: First-order theories of accounting derived from conceptual (i.e. semantic) analysis of accounting practice and accounting theorists, Reconstructive expertise of accounting system implementers largely derived from textbook descriptions of 'typical' bookkeeping systems, and implementation heuristics for construction of events-based accounting systems derived from the database design experiences of the authors in such work (Rockwell and McCarthy 1999, 182).
Geerts and McCarthy (2000a)	CREASY	CREASY was a CASE tool written in PROLOG that supported the development of accounting systems with intensional reasoning capabilities.
Chen and McLeod (1995)	REAtool	REAtool was a CASE implementation designed to support the evolution of REA schemas in accordance with heuristics developed by Batini, Ceri, and Navathe (1992)
Dunn (1994)	REA abstraction interface	This was a SMALLTALK tool that built a semantic abstraction interface based on REA models on top of a relational database
Cherrington et al. (1994)	IBM payroll (production system)	This implementation was a payroll system developed at IBM using REA accounting principles
Walker and Denna (1997)	Price-Waterhouse *GENEVA* system	*GENEVA* (GENeralized EVents Architecture) is a tool based on REA principles that was developed by Price-Waterhouse to support fast retrievals from large databases
Haugen (1997), Haugen and McCarthy (2000)	*Supply Links* Package	*Supply Links* is a package designed with REA principles that optimizes the synchronization of dependent demand across multiple partners in an integrated supply chain.