

Equational zero vector databases, non-equational databases, and inherent internal control

Roberta Ann Barra

College of Business and Economics,
University of Hawai'i at Hilo,
200 West Kawili St., Hilo, HI 96720, USA
E-mail: Roberta.barra@hawaii.edu

Arline Savage

Orfalea College of Business,
California Polytechnic State University,
1 Grand Avenue, San Luis Obispo, CA 93407, USA
E-mail: savage@calpoly.edu

Jeff J. Tsay

College of Business,
University of Texas at Arlington
701 S. West Street, Arlington, Texas 76019-0468, USA
E-mail: jefftsay@uta.edu

Abstract: Equational zero vector accounting systems, based on duality principles and the double-entry model, were designed as ontological control systems to help prevent and detect fraud and errors inherent in non-equational, single-entry systems. Non-equational systems lend themselves to fraud and errors to a larger degree because the internal control inherent in an equational zero vector system has no substitute. We use an analytical analysis methodology to show that an equational zero vector system provides superior inherent internal control over data completeness and data reliability. In the accounting information systems area, the most popular modern non-equational system, the resource-event-agent model, is increasingly being promoted as a replacement for the equational zero vector accounting system. We contend that, although non-equational accounting system frameworks can be modelled with controls, they do not achieve the degree of control inherent in an equational zero vector accounting system without becoming an equational zero vector accounting system.

Keywords: accounting information systems; AIS; non-equational databases; COBIT; double entry; equational zero vector databases; internal control; resource-event-agent; REA.

Biographical notes: Roberta Barra is an Assistant Professor of Accounting at the University of Hawai'i at Hilo. She has had many years of industrial teaching and research experience in auditing, systems and software consulting involving accounting and ERP software. Her research has focused on accounting information systems including internal controls, accounting software evaluations and documentation techniques. Her secondary interests include international accounting and audit-related research. She has presented her research at conference proceedings nationally and internationally. She earned an MBA from the University of Houston and a PhD from the University of Illinois at Urbana-Champaign.

Arline Savage is a Professor of Accounting and Deloitte Faculty Fellow at Cal Poly San Luis Obispo, California. She has taught accounting information systems (AIS), systems analysis and design, IT audit, business process analysis, and fraud examination. Her research interests include controls in information systems. She is the past President of the AIS Educator Association and was the Founding Editor of its academic journal. She earned her doctorate from the University of Port Elizabeth (now Nelson Mandela Metropolitan University) in South Africa.

Jeffrey J. Tsay is an Associate Professor of Accounting at the University of Texas at Arlington. He earned his PhD from the University of Missouri-Columbia. He has authored a textbook in Visual Basic programming and has published articles in the *Journal of Accounting Research*, *The Accounting Review*, *Journal of Management Accounting Research*, *the Journal of Financial Research* and many other journals. He mainly teaches accounting information systems.

1 Introduction

For over a quarter of a century, the extant accounting information systems (AIS) literature (e.g., Geerts and McCarthy, 2002; McCarthy, 1982; Rockwell and McCarthy, 1999; Verdaasdonk, 2003) has, with increasing frequency, been calling for the demise of double-entry accounting. This literature offers a multiple-component, single-entry model, the resource-event-agent (REA) ontology as the alternative to double-entry database systems. The REA ontology "...abandons debits, credits, and traditional account structures as artifacts associated with the mechanics of journals and ledgers..." [Church and Smith, (2007), p.2].

While largely ignored by auditing and financial accountants, the REA model has enormously impacted the AIS area. Most AIS textbooks focus heavily on the REA approach [Batra and Sin, (2008), p.200], with many academics working on the development of new accounting systems [Kim, (2009), p.174]. Recently, the International Organization for Standardization approved the open electronic data interchange (EDI) standard which provides an explicit REA specification of an accounting and economic ontology to support business transactions among collaborating partners (International Organization for Standardization, 2007). Additionally, the National Science Foundation (NSF) granted funds to explore the establishment of a domain ontology standard for financial accounting software (NSF, 2007). For the academic delegation, it appears that only REA-supporting participants were invited to this NSF grant-supported workshop.

For example, one of the REA proponents made this personal statement on the workshop website:

“My primary research interest with respect to REA is its use in the development of improved financial and managerial reporting models including revised financial statements, standards, and executive dashboard elements. Independent-view reporting is needed and requires a solid ontological foundation. Many existing assumptions and principles (such as periodicity, conservatism, matching, absorption-costing, and others) must be replaced.” [Dunn, (2008), slide 2 of 25].

AIS textbooks also predict the demise of double entry accounting. A case in point is Romney and Steinbart (2009), one of the market leaders:

“Database systems may profoundly affect the fundamental nature of accounting. For instance, database systems may lead to the abandonment of the double-entry accounting model. ...If the amounts associated with a transaction are entered into a database system correctly, then it is necessary to store them only once, not twice. Computer data processing is sufficiently accurate to make unnecessary the elaborate system of checks and double checks that characterises the double-entry accounting model.” [Romney and Steinbart, (2009), pp.123–124].

Contrary to this position, this paper is a defence of double-entry accounting in modern AIS. While some describe double-entry accounting as artifactual, expendable, unappealing and myopic [e.g., Batra and Sin, (2008), p.200; Church and Smith, (2007), p.2], we provide mathematical proof to refute this position. We argue that the mechanics of ledgers and journals are not artefacts, but are instead an integral part of the inherent internal control structure embedded in an accounting system.

We agree with Batra and Sin (2008, p.200) that the REA approach denotes only a data model, which is the static aspect. It does not include the dynamic aspect of how data is processed, which is central to any AIS. To date, we know of no fully-functioning REA enterprise system, although models exist for all the component parts [e.g., Church and Smith, (2007), p.7; Hesselund, 2006). REA, in and of itself, does not stand alone. Unlike the double-entry system, it needs to be extended to become a fully-functioning accounting system [e.g., Batra and Sin, (2008); Sedbrook and Newmark, 2008]. It has to increase in complexity to accommodate new complexities in business, which in turn makes it less auditable.

In contrast, the double-entry system has stood the test of time since the 13th century. It is the basis for billions of working information systems that are auditable and that have been able to adapt to the complexity of the sophisticated business transactions in use today. It is a flexible, adaptable and incredibly sophisticated in its mathematical simplicity. Consequently, our research goal is to show that use of a double-entry database system provides superior inherent internal control for detecting and preventing errors over data completeness and data reliability than a single-entry database system such as REA.

The remainder of this paper is organised as follows. In the next section, we discuss the background for the research. This is followed by our research methodology. The fourth section investigates equational zero vector accounting systems (EZS) and non-equational accounting information system (NES) as competing ontologies of database accounting systems, which shows that EZS does meet the criteria set by the NES proponents. In the penultimate section, we provide analytical proof that EZS is

superior to NES from an inherent internal control standpoint. In the final section, we present our discussion and conclusions.

2 Background for the research

Auditors have long been interested in the impact of internal control on system design (e.g., Alali et al., 2008; Bodnar, 1975; Borthick, 2008; Cooley and Cooley, 1982; Cushing, 1974; Fields et al., 1986; Grimlund, 1982; IT Governance Institute, 2006; Knechel, 1983; Srinidhi, 1988; Weber, 1980; Yu and Neter, 1973) because of their reliance on these controls in the performance of the audit. Now, with Sarbanes-Oxley (2002) legislation (SOX), auditors must attest to management's assessment of the effectiveness of internal controls over financial reporting (i.e., see Kumar et al., 2008). Given that the vast majority of financial reporting today is performed in a computerised environment, any inherent control risks of the database design become all the more important. Moreover, Sutton and Arnold (2005, p.125) make the case that the "CIO will be inextricably linked to the SOX Act efforts...and (will have) little choice other than to become a key participant in the ... risk management process—and seemingly a leader in (this) area." The growing participation of IS professionals in the risk management process is demonstrated by Ho et al. (2009) and Javidi et al. (2009). Consequently, this research has important implications for auditors, financial accountants, chief information officers and system designers.

If one ignores higher order models, there are two choices one can use when designing a computerised AIS. One can design an equational zero vector (or double entry) accounting information system (assets – liabilities – equities = 0). Or, one can design a NES. There are many possible non-equational systems, from old ship master's logs in ancient times to the popular REA system of modern time. In this analytical investigation, we show that use of an equational zero vector system provides superior inherent internal control for detecting and preventing errors over data completeness and data reliability. We contend that, although non-equational accounting system frameworks can be modelled with controls, they do not achieve the degree of control inherent in an EZS without becoming an EZS.

We begin by comparing EZS and NES as though they were competing ontological models of database accounting systems. We do this because the extant AIS literature (e.g., Verdaasdonk, 2003; Geerts and McCarthy, 2002; Rockwell and McCarthy, 1999) treats them so. Our position is that EZS and NES are not competing ontological models of database accounting systems. Rather, they are complimentary to each other. NES can be used to model EZS if one chooses, but EZS should be the system modelled.

Manual EZS had been in use, side by side, with manual NES for centuries. Then, more than a century ago, manual EZS replaced manual NES definitively because of the inherent control deficiencies of NES (Ashworth, 1886; Barra and Griggs, 2007; Bardsley, 1900; Craig, 1898; Marshall, 1895). A computerised environment has not solved these inherent control deficiencies and the financial accounting and auditing profession has consequently largely ignored computerised NES (McCarthy, 2003; Poels et al., 2007). Just as EZS solved certain internal control deficiencies of NES in manual systems, it continues to solve those NES deficiencies in a computerised environment.

We concur with the position stated in Geerts and McCarthy (2002, p.4) that "any domain ontology...be subject to peer review not only in computer science or knowledge

representation fields, but additionally in applied disciplines as well.” Given this standard, NES fails the test to be useful as an application because it fails to meet the auditing and legal requirements of internal control. With this research, we show that the inherent internal control deficiencies of NES, such as the REA model proposed by McCarthy (1982), make these models unsuitable as AIS applications under the current standards of COBIT, the Committee of Sponsoring Organizations (COSO, 1992), the Public Company Accounting Oversight Board (PCAOB), and SOX.

Nonetheless, modern NES such as REA have proven useful as modelling tools for building databases for business (e.g., Bradford et al., 2007; Callaghan et al., 2002; Dunn et al., 2005; Gerard, 2005; Jones et al., 2002, 2005–2006; Poels et al., 2007), including non-accounting databases (e.g., Vymetal et al., 2008). Consequently, there is room for both NES and EZS in a modern AIS: one as a database design tool, the other as the target of that design. It should be noted, though, that unified/universal modelling language (UML) is the standard for database design in the management information systems and software engineering literature (Dominguez et al., 2002; Joosten and Puro, 2002; Larsen et al., 2007; Selic, 2006). Proponents of NES as a database design tool would have to defend their models against the UML models. This has been done somewhat successfully in prior research (Jones et al., 2002, 2005–2006), although this research was preliminary in nature.

3 Research methodology

We use an analytical approach for this investigation because this methodology is the most suitable as it provides the most generalisable results. Analytical studies can explore the nature of internal control, comparing the most general of cases between EZS and NES. While simplifying assumptions do limit the generalisability of analytic research, to the extent the assumptions are valid and/or reasonably close to the actual state of nature, generalisability is increased. In this study, we build our analytics with a minimum of assumptions, using instead lemmas from algebra, accounting and auditing. This further increases the generalisability of the results.

For an experimental study, the researcher would necessarily have to test both types (EZS and NES) of systems internal control efficacy against fraud as well as errors. This would require the use of some monetary incentives. Such monetary incentives in an experimental setting would be limited to a relatively small amount for each test subject. This would probably not be a problem for studying errors. However, for studying the impact of internal control on fraud, it would be analogous to petty theft. Controlling petty theft may be fundamentally different from controlling major fraud. Such a fundamental difference, if it exists, would limit the external validity of an experimental study.

An empirical study is limited in that a researcher can only study those systems that have been fully developed and those frauds and errors that have been both detected and reported. To our knowledge, there exists an insufficient number of full-functioned NES accounting systems to lend statistical validity to even non-parametric testing. Moreover, there are certain to be, in the short term at least, undetected frauds and errors. There are also detected frauds and errors that are never reported, but are handled internally by organisations. Thus, only a subset of the total population of fraud is available for research. More importantly, undetected frauds and errors, along with detected but unreported frauds and errors, may be fundamentally different from detected and reported

frauds and errors. Again, such a difference, if it exists, would limit the external validity of an empirical study.

Analytical research can provide theories to be tested by those doing quality empirical and experimental work. The theory informs the whole process of data collection and can help strengthen the results of experimental and empirical work. It is for these reasons that we take an analytic approach.

4 NES and EZS as competing ontologies of database accounting systems

While practising auditors and financial accountants may think it perplexing to consider anything other than an EZS, many in the AIS academic community have been designing and advocating NES since McCarthy's (1982) publication. A segment within this community, starting with McCarthy (1982), has gone so far as to argue that EZS should be abandoned because it is deficient (e.g., Geerts and McCarthy, 2002; Rockwell and McCarthy, 1999; Verdaasdonk, 2003). Consequently, we think it is important for rigor to address the deficits of EZS as claimed by some REA proponents.

Theorem 1: The EZS model has the same three core features as modern NES models, which are:

- 1 a database orientation
- 2 a semantic orientation
- 3 a structuring orientation.

Lemma 1: One of the three core features of modern NES is a database orientation. According to Dunn and McCarthy (1997), this means having the following three characteristics:

- 1 data are stored at their most primitive level, at least for awhile
- 2 data is stored such that all authorised decision makers have access to it
- 3 data is stored such that it may be retrieved in various formats.

None of the required database orientation characteristics are inherent in the NES itself, but it can be modelled to have these characteristics. For example, there is nothing in REA to imply that data is stored such that it may be retrieved in various formats. An analyst would need to consider how best to record the business events and then use a process of normalisation, through third normal form for most applications, in order to accomplish storing a REA model at their most primitive level and allowing that data to be retrieved in various formats (Callaghan et al., 2002). An improperly normalised REA model could still model resources, events and agents, yet not be at the most primitive level and not allow data to be retrieved in various formats. This is a function of the database analysts' design. Consequently, we submit that any accounting model, including an EZS, can be modelled to have a database orientation as it is defined above.

Certainly, the characteristics presented in numbers 2 and 3 (i.e., data are stored at their most primitive level, at least for awhile and data is stored such that all authorised decision makers have access to it) are common to virtually all financial accounting software packages using the double-entry model. The empirical evidence supports this as

virtually all packaged financial software today using the EZS model also use a relational database or an object-oriented database (Jones, 2000, 2002, 2007). We argue that the same is true for characteristic number 1 (i.e., data is stored at their most primitive level, at least for a while), and that this is common to many financial accounting software packages using EZS. However, in the REA literature, strawman arguments have been presented with respect to number one, which requires further discussion. Dunn and McCarthy (1997) state the following while discussing semantics: "...components of the model should reflect real world phenomena, a situation that precludes the use of basic double-entry artefacts (e.g., debits, credits, accounts) as *declarative primitives*" (emphasis added). In computer science, declarations are essentially 'name = value' pairs. When the value is a primitive (integer, Boolean, double or string) then one has a declarative primitive. For example, specifying that the 'social security multiplier equals 0.0765' is a declarative primitive, with the field name equalling 'social security multiplier' and the value equalling '0.0765'. We would be hard-pressed to think of many examples in an EZS or NES that would be declarative primitives. Fortunately, the database requirement does not require declarative primitives, but merely that data be stored at their most primitive level, namely, data should not be the result of calculations. This would not preclude dates, for example, given the language used by the cited authors, because clearly a date is at its most primitive level.

Therefore, EZS models can be modelled with a database orientation.

Lemma 2: The second of the three core features of a modern NES is a semantic orientation. According to Dunn and McCarthy (1997), 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." Semantics is the study of meaning. In computer science 'Let $X = X + Y$ ' and ' $X = X + Y$ ' have different syntax (different computer language) but the same semantics (meaning.)

Prior research (Dunn and Grabski, 2000) has shown that EZS is semantically expressive. The research finds that the particular NES, REA, is more semantically expressive than EZS. The REA system had a mean of 5.329 (on a scale of 1 to 7) and the EZS system had a mean of 4.164. This research indicates some degree of economic semantic expressiveness in the EZS model. It is not our goal to prove that EZS is more semantically expressive than NES, merely that EZS are semantically expressive. This prior research accomplishes that task.

We go further to explain the dual nature of the semantic expressiveness inherent in an EZS model. We show that the EZS model is a translation of the economic semantic into the mathematic semantic and retains elements of both.

The basic accounting equation ($A = L + E$; $A - L - E = 0$) is a real equation and retains all the qualities of an equation. Hence, double-entry accounting is applied algebra, which is its language. The semantics, the meaning of mathematics, is universally understood. Because of this universality, anyone who understands mathematics can understand and use the information in double entry accounting. This has been demonstrated by research in economics, financial accounting and physics that take the double-entry model as their starting point (Ashton et al., 2004; Braun, 2001; Fischer and Braun, 2003; Lundholm, 1995).

Business events must be translated into this equation, but all accounting equations in an EZS are also encoded with account titles that represent the economic resource (e.g., cash, inventory, fixed assets and investments) and/or its control account (e.g.,

accounts receivable, accounts payable and inventory). A standard journal entry has the economic encoding of the resources owned or owed by the firm, the equity change resulting from the transaction, the dollar amounts involved, all in accordance with generally accepted accounting principles (GAAP) and all presented such that the equational zero vector inherently maintains the zero balance. If there is a valid criticism of EZS, then it is in this translation. Because a translation is always required of the accountant (i.e., transaction analysis), one must translate the real world phenomenon from one language (business transactions, i.e., economic semantics) into two other languages: GAAP and mathematics (mathematic transactions, i.e., difference semantics in our case – see Mattessich (1995, pp.86, 92–94) for theorem and proof). A journal entry in an EZS inherently has all these semantic components. The real issue here is the assertion of some NES proponents that EZS does not reflect real-world phenomena. This comes from two sources. The first source arises from labelling the control portions of EZS as artefacts. Debits, credits, the journals, ledgers and sub-ledgers are all a part of the control function of the EZS. This will be discussed in more detail in a later section of the paper.

The primitives in EZS are the accounts and sub-accounts upon which vector operations (see below) are performed. These accounts do measure real-world phenomena: cash, inventory, fixed assets, marketable securities, stock, Customer A, Vendor B, (which could all be found in the subsidiary accounts) and so forth. Then, as part of the control function, we have additional control accounts, some of the most common being accounts receivable, accounts payable, inventory and fixed assets. These primitives are so similar to the *resources* and *agents* found in the modern REA NES that they would make no material difference to an auditor and should make no material difference to a database designer.

The second source of EZS opponents' assertion that the double-entry accounting model does not reflect real-world phenomenon comes from the failure of EZS to record non-financial information. This results from the failure of EZS to do so in the past; not from any inherent inability in the model to accommodate non-financial information. One can just as easily encode non-financial information in the form of debits/credits as one can encode financial information. Think in terms of introducing duplicity into a modern NES model and you begin to see the proof of this. A formal proof is presented in Ellerman (forthcoming – see Appendix A). An example is presented in Appendix B.

It is inaccurate to say that EZS does not reflect financial real-world phenomenon. The NES proponents may be correct that a good accounting model should reflect all real world phenomena, including the non-financial information components. If this is so, they have every right to chide EZS proponents for their failure to finish the job (McCarthy, 1982; Rockwell and McCarthy, 1999). Indeed, it has been shown in the financial empirical literature that financial information alone can be largely irrelevant; that it is only when such information is combined with industry-specific non-financial information that such information explains stock prices (Amir and Lev, 1996). But it is overstretching to infer '*can not*' from '*have not*' in the case of the non-financial component.

Moreover, while the financial accounting community itself has yet to incorporate non-financial information into the EZS model per se, database designers of EZS application software have not overlooked non-financial information in their application designs, nor have managerial accountants omitted the concept of responsibility accounting. In today's modern software applications, even at the low end (e.g., PeachTree), the financial accounting portion of the available functionality is

decreasing as software vendors add more and more functionality to their product lines (O’Leary, 2004; Jones, 2000, 2002, 2007). It may well be the case that the issue of recording and retrieving non-financial information has already been solved by software vendors, without resorting to NES models, and that this has become a moot point.

Therefore, double-entry accounting has a semantic orientation that records financial real-world phenomenon and can accommodate recording non-financial real-world phenomenon.

Lemma 3: The third core feature of a modern NES is a structuring orientation or the repeated use of an occurrence template as a foundation or accountability infrastructure for the integrated business information system.

The best known EZS model, the debit/credit, double-entry accounting system, has a well established system of accounts with journals, ledgers and sub-ledgers. Moreover, this structure in an EZS system is almost fractal in nature in that it is self-similar throughout the system. Every process and sub-process within an EZS system has an equational zero vector. If one models the entire system, an entire process of the system, or one very small process within another process, one knows that the accounts will all have an equational zero vector. They will all be self-similar.

Therefore, the EZS model has a structuring orientation.

Therefore, the EZS model has the same three core features as modern NES. QED

5 EZS is superior to NES from an inherent internal control perspective

It is important that control be designed into a system. When control is inherent to a system and inescapably part of the system design, then that system must be superior from an inherent control perspective compared to a system where this is not the case. According to Solari (2007, p.1): “The time to design security into ... technology is ... before we find that the systems are deployed, operational, and optimised with a degree of complexity that simply cannot be adequately remedied by retrofitting security after the fact...Retrofitting security...as a strategy for risk management...is largely ineffective in lowering overall risk as the remediation can only serve as a stop gap.” In this section, we show here that EZS is superior to NES from an inherent internal control perspective.

Theorem 2: From an inherent control standpoint, the EZS model is superior to any NES.

Lemma 4: COBIT is the accepted standard for IT governance.

Lemma 5: Control is one of the core elements of IT governance, according to the COBIT framework.

Lemma 6: COBIT requires that the control framework over IT fit with and support the COSO control framework.

The COSO framework has reliability of financial reporting as one of the three objectives of internal control. Compliance with laws and regulations is another objective of internal control which implies compliance with SOX legislation. Reliability of financial reporting is defined by the COSO framework to be fairly presented in conformity with GAAP; supporting fair presentation are the five basic financial statement assertions: existence or

occurrence, completeness, rights and obligations, valuation or allocations, and presentation and disclosure.

Moreover, according to the COSO, a reportable condition is an internal control deficiency related to financial reporting; it is a significant deficiency in the design or operation of the internal control system that could adversely affect the entity's ability to record, process, summarise and report financial data consistent with the assertions of management in the financial statements.

Lemma 7: "COBIT's good practices represent the consensus of experts. They are strongly focused more on control, less on execution" [COBIT 4.1 ,(2007), p.5].

Therefore, one of the goals of an automated AIS must be control.

Lemma 8: COBIT's information criteria include, among others:

- 1 integrity, accuracy and completeness
- 2 reliability.

(Note that COBIT distinguishes among these criteria. In the discussion that follows we will argue that EZS are superior to NES for only two of these criteria: completeness and reliability. We do not make similar claims for the criteria of integrity or accuracy).

Lemma 9: Controls that should be embedded in business process applications (application controls) include completeness checks and controls to ensure that all transactions are accurate, complete and valid [COBIT 4.1 ,(2007), pp.15–16] (emphasis added).

Lemma 10: SOX Section 103.A.2.A.iii.2.aa requires that internal control structures and procedures include maintenance of records that in reasonable detail accurately and fairly reflect the transactions and deposits of the assets of the issuers.

Lemma 11: SOX Section 103. A.2.A.iii requires auditing standards...shall describe in each audit report the scope of the auditor's testing of the internal control structure and procedures of the issuer.

Therefore, automated AIS must have controls that are auditable.

Lemma 12: Transaction data that are deleted or missing are incomplete data.

Lemma 13: Incomplete data are unreliable data.

Lemma 14: Incomplete and unreliable data are at least poorly controlled data.

Lemma 15: Modern NES is a multiple component single-entry system that relies on a non-mathematic semantic orientation.

Lemma 16: An EZS database is a multiple component double-entry system that relies on an economic/GAAP/mathematic semantic orientation (the group of differences).

Given the above lemmas, we propose the following hypotheses:

- Hypothesis 1A An equational zero vector automated accounting system provides better control over data completeness than a non-equational automated accounting system.

- Hypothesis 1B An equational zero vector automated accounting system provides superior control over data reliability than a non-equational automated accounting system.
- Hypothesis 2 An equational zero vector automated accounting system is less risky to audit than non-equational automated accounting system.

Proof of Hypothesis 1A, Part 1

An equational zero vector system has a *fourth* core component that is not a component of a NES. This component is the control component inherent in the balancing nature of any algebraic equation which holds, if, and only if, the two sides of the equation maintain their equational zero account. Ellerman (1982, 1985, 1986, forthcoming) demonstrates that (see Appendix A):

$$\begin{aligned} & \text{Beginning Equational Zero-Accounts} \\ & + \text{Transactional Zero-Accounts} \\ & = \text{Ending Equational Zero-Accounts} \end{aligned}$$

or in more conventional terminology,

$$\begin{aligned} & \text{Beginning General Ledger} \\ & + \text{Journal} \\ & = \text{Ending General Ledger} \end{aligned}$$

Proof of Hypothesis 1A, Part 2

We have shown from Proof, Part 1, that when we move from one time period to the next with an EZS, the following is true:

Assets – Liabilities – Equities = 0 at the beginning of the period

Add the current transactions such that Assets – Liabilities – Equities = 0

Then the ending balances will also hold such that Assets – Liabilities – Equities = 0.

So there exists only one state of being: zero. If the system is not in this state then we know that there is an error or irregularity in the system.

Now consider a nonequational system from the set of real numbers (i.e., where negative numbers, zero and positive numbers are possible). Then such a system can be described as follows: The resources R at the beginning of the period, denoted as R_{T_0} , plus resources added during the period, less resources used up during the period, denoted as R_t , equals the resources at the end of the period, denoted as R_{T_0+1} .

The resources come from the set of real numbers and individual transactions can be either positive or negative. For example, a purchase transaction would be a positive transaction. A cash disbursement would result in a negative transaction. The summation of all resources at the beginning of the period, while one hopes would net to a positive number, can conceivably be an element of the set of real numbers. The same is true for the summation of all the transactions for the period. So we have the following:

$$\sum_{i=1}^n R_{T_0} + \sum_{j=1}^m R_T = \sum_{k=1}^p R_{T_0+1}$$

The following cases are possible:

Case 1A, 1B and 1C:

$$\text{if } \sum_{j=1}^m R_T = 0$$

$$\text{then } \sum_{i=1}^n R_{T_0} = \sum_{k=1}^p R_{T_0+1} \quad (1)$$

Case 1A When $\sum_{i=1}^n R_{T_0} = 0$ as well, then this is the special case of the equational zero vector system. This case always holds for an equational zero vector system when there are no errors in the system. For the equational zero vector system, the inherent control is that a non-zero state *always* indicates a state of error in the system.

Case 1B When $\sum_{i=1}^n R_{T_0} = 0$ and $\sum_{i=1}^m R_T > 0$ then $\sum_{k=1}^p R_{T_0+1}$ should be positive. (2)

Case 1C When $\sum_{i=1}^n R_{T_0} = 0$ and $\sum_{j=1}^m R_T < 0$ then $\sum_{k=1}^p R_{T_0+1}$ should be negative. (3)

Case 2A and 2B:

$$\text{if } \sum_{j=1}^m R_T < 0$$

$$\text{then } \sum_{i=1}^n R_{T_0} > \sum_{k=1}^p R_{T_0+1} \text{ if } \sum_{i=1}^n R_{T_0} > 0 \quad (4)$$

$$\text{and } \sum_{k=1}^p R_{T_0+1} < 0 \text{ if } \sum_{i=1}^n R_{T_0} < 0 \quad (5)$$

Note that in Case 2A equation (4), however, it is not possible to know whether R_{T_0+1} is positive or negative; it is dependent on the relative size of R_i and R_{T_0} .

Case 3A and 3B:

$$\text{if } \sum_{j=1}^m R_T > 0$$

$$\text{then } \sum_{i=1}^n R_{T_0} < \sum_{k=1}^p R_{T_0+1} \text{ if } \sum_{i=1}^n R_{T_0} < 0 \quad (6)$$

$$\text{and } \sum_{k=1}^p R_{T_0+1} > \sum_{i=1}^n R_{T_0} > 0 \text{ if } \sum_{i=1}^n R_{T_0} > 0 \quad (7)$$

Note that in Case 3A equation (6), however, it is not possible to know whether is positive or negative; it is dependent on the relative size of R_t and R_{T_0} .

Thus, in the nonequational system there are multiple system states that can ensue and devolve into three possible end states: R_{T_0+1} can be positive, negative or zero; the entire set of real numbers. For only one of these states do we know the value: Case 1A, in which the value of the state is zero. There are two states [equations (4) and (6)] in which we do not know the sign, but we do know the size of ending account balances relative to beginning account balances. There are four states in which we know the signs with certainty, equations (2), (3), (5) and equation (7); two of these states are positive, the other two are negative. Thus, in a NES the current state of a system encompasses the entire set of real numbers.

The issue is not what the current state of the system is, but what the current state of the system should be. In the equational zero vector system, we know that the system state should be zero, always. If the state is not zero we know the state is in error, always. This does not ensure that all transactions have been recorded or are valid. The control inherent in an EZS is inherent, not sufficient, for an AIS. Even an audit, which is designed to test controls and attest to the fairness of the financial statements, cannot make assurances about the absolute certainty that all transactions have been recorded and that all transactions are valid. But what one *can* say with *absolute certainty* is that if an EZS system state is not found to be zero then there *is* an error in the system.

In the nonequational system, one cannot know in which state the system *should be* from the system itself. One merely knows in which state one finds the system; not what the system should be. In a NES, one knows nothing with absolute certainty.

Proof of Hypothesis 1A, Part 3

The beginning and ending balances of the general ledger are nothing more than the equational zero vector of the EZS at two different points in time. The detail in the general ledger also provides the transaction vectors that allow one to move through time from one equational zero vector to the next equational zero vector. These equational zero vectors, at two different points in time, with the transaction vectors that move one through time, were designed to provide an audit trail to help prevent or detect errors and irregularities. The general ledger and the general journal are not artefacts of the EZS; rather, these are the proofs of the EZS. Consider the following:

$$3X + 2 = 14$$

$$3X = 14 - 2$$

$$3X = 12$$

$$X = 4$$

The first equation is equivalent to the last equation. The middle equations merely serve to prove how we progress from the first to the last. In the same way, a beginning general ledger is an equational zero vector (proofed via trial balance in the old manual systems) and the ending general ledger is also an equational zero vector (also proofed via trial balance), thus mathematically equivalent to the beginning general ledger even though the individual account balances may differ. Indeed, all general ledgers are mathematically equivalent in an EZS. The journal entries are the evidential path themselves, encoded as equational zero vectors, which take one from the beginning balance to the ending balance, thus providing the audit trail and the essential control inherent in the EZS. This control feature is inherent in an accounting EZS because such a system relies on maintaining the integrity of a mathematical equation; i.e., keeping the equation in balance. A NES, however well conceived, has no such equation to balance and must, inherently, lack this control.

In modern automated EZS, some generate the journals and ledgers via a reporting system. These systems treat each individual account as an equational zero vector and use programming to perform the mathematical translation and reports required to generate the journals and ledgers. But even the most extreme of these systems retain the equational zero vector tests (i.e., assuring that each transaction, each account, equates to zero; rather trivial computer operations yet extraordinary controls), journal and ledger reports, and the general ledger control accounts to provide the control features required by internal and external auditors implementing COSO, COBIT, PCAOB and SOX requirements.

Proof of Hypothesis 1A, Part 4

The general ledger includes control accounts, such as accounts receivable and accounts payable, which can be supported by subsidiary ledgers. These dedicated control accounts often do not exist in a NES. Rather, they are generated via calculated query as the difference between sales and cash receipts (Klamm and Weidenmier, 2004; McCarthy, 1982, 2003; Rockwell and McCarthy, 1999). The problem with generating accounts receivable by subtracting cash receipts from sales is that this does not give one accounts receivable. This gives one accounts receivables netted with unearned revenue. In order to obtain accounts receivable, one must either redesign the NES to somehow tag unearned receipts with additional attributes to make them stand out when querying the database or one must analyse each cash receipt/sales transaction to determine whether it belongs in the asset account or the liability account. Similarly, subtracting cash disbursements from purchases gives one accounts payables netted with prepaid expenses. McCarthy (1982) made this clear but this fine point is, unfortunately, overlooked in subsequent research.

The control accounts (accounts receivable, accounts payable, fixed assets, etc.) backed by their subsidiary ledgers, provide not only useful information but also provide a powerful reconciliation control used by external auditors, in-house accountants and internal auditors. The redundancy of recording totals to the control accounts and detail transactions to the subsidiary ledgers makes it more difficult for errors and irregularities to go undetected during the reconciliation process. Because NES often lack both the control accounts and these redundancies, these reconciliations are not possible. A NES database can calculate an *accounts receivable* balance. But without these reconciliations and zero vector control over data input, a NES database cannot provide reasonable assurance that the amount calculated is the correct amount without a transaction-by-transaction analysis to determine whether *accounts receivable* are

complete and which cash receipts belong in *unearned revenue*. Similar logic is true for *accounts payable* and *prepaid expenses*. This constitutes a material weakness in internal control, and is a reportable condition.

Proof of Hypothesis 1A, Part 5

Proof is by counter example. Suppose that an automated NES provided superior control over data completeness when compared to an automated EZS.

The NES framework could be modified in at least one of two ways to introduce a control element. First, a modelling construct could be introduced to incorporate control. Alternatively, one might generalise the controls features of the EZS, to the extent possible while maintaining the essential nature of a NES as a single-entry system (otherwise one evolves into an EZS), and then integrate these into the NES.

Now, suppose that by adding these controls, the NES was at least as good as EZS at controlling transactions, and yet remained essentially a NES. Consider a recorded transaction in which only two assets (two resources) are increased, with no other resources affected by this economic event.¹ It is apparent to any accountant who knows double entry that this cannot possibly be correct. Consequently, in the EZS it would be immediately apparent that this entry was incorrect. In a system with hundreds of accounts/resources, the thought required to detect potential errors of this kind (increase both a liability and an equity, decrease two assets, and so forth) would increase exponentially in an NES, however well controlled via modifications to the NES framework, but this issue remains trivial in a EZS system where the control is inherent to the framework.

Errors of this kind, where part of the transaction is missing and/or the transaction is incompletely specified, can happen in an NES framework because it is a single-entry system; especially when one considers that relatively unskilled clerks perform most of the initial data entry into any accounting system. The EZS use of equational zero accounts provides for an immediate control over the plausibility of a transaction. Thus, the counter example shows that any NES model does not provide more control than the EZS over data completeness.

Therefore, Hypothesis 1A is supported; an equational zero vector automated accounting system provides better control over data completeness than a non-equational automated accounting system. QED

Proof of Hypothesis 1B, Part 1

Proof is again by counter example. Suppose that an automated NES provides superior control over data reliability than an automated EZS. Input a transaction into an NES with a transposition error. In a financial transaction, this error might be detected while performing the bank reconciliation. If it is non-financial, for example inventory unit data, this might be detected while performing the inventory count at year-end. In an EZS system, a transposition error will be detected immediately if only one side of the entry is transposed. If both sides are transposed, then the error will be detected while reconciling the subsidiary ledgers to the control accounts in the general ledgers. (An NES can also provide ledger balances, through database query, for reconciliation but these balances would be netted with either unearned revenue or prepaid expenses, making reconciliation

much more difficult, if not impossible.) If not detected in this manner, then the transposition will be detected in the same manner as in the NES model. Thus, the EZS model will detect transposition errors at least as soon as the NES model or sooner than the NES model. This counter example shows that the NES model cannot provide superior control over data reliability.

Proof of Hypothesis 1B, Part 2

Certain of the so-called 'artefact's EZS, such as the debits/credits and control accounts coupled with their attendant subsidiary ledgers, were designed as built-in redundancies, much as the space shuttle and our computers have built-in redundancies designed to protect, in this case, the firm from malfeasance and error (Ashworth, 1886; Bardsley, 1900; Craig, 1898; Marshall, 1895). Although initially designed for protection against manual error and fraud errors, this does not make these redundancies any less relevant in computerised environments.

It has been shown in the reliability literature (e.g., Bodnar, 1975) that, *ceteris paribus*, the reliability of a system with built-in redundancy is greater than or equal to the reliability of a system with non-redundant components. Redundancies were also considered necessary by Hammer and McLeod (1981) to "simplify user's manipulation of a database by statically embedding in the schema data values that would otherwise have to be dynamically and repeatedly computed" (Hammer and McLeod, 1981).

The specialty journals and the subsidiary ledgers exist to serve as control tools (Cushing, 1974). This is their function. To label them artefacts is to diminish their very important function. Specialty journals serve to ensure that transaction vectors, both the debit side and the credit side, are uniformly encoded. Subsidiary ledgers also help to ensure uniform encoding of transaction vectors, but furthermore serve to relieve some of the cognitive pressures involved in working with a large number of accounts; thus they control cognitive errors. Subsidiary ledger accounts are also a lower level of aggregation along a desired/desirable dimension of efficiency. Individual accounts receivable or accounts payable account balances are needed for daily operations. It is plainly inefficient to query the database (on all data included from its inception) each time a lookup is required. Thus, subsidiary ledgers are not only useful as control tools but are also efficient.

Subsidiary ledgers, while control tools, also contain entity information considered *agents* or *resources* in certain NES models: customers, vendors, inventory items, fixed asset items and so forth. As useful as specialty journals and subsidiary ledgers are in any financial accounting system as control devices and to increase efficiency, they have always been optional. The information encoded in these journals and ledgers can be encoded in a general ledger and a general journal if a database designer chooses; they are not inherently required by an EZS but have a long history of use because of their value. (Beginning balances in subsidiary accounts are an important redundancy to ensure accuracy. If one of these balances is altered subsequently, the entire set of accounts becomes out of balance and is easy to detect. On the other hand, with an NES system, there is no easy way to detect a change to a prior period transaction that results in a change in an account balance since no such redundancy is in place.)

Therefore, Hypothesis 1B is supported; for control over data reliability, an automated EZS is superior to any automated NES model. QED

Given that from a control standpoint the double-entry accounting model is superior to the NES accounting model, then the control risk of the EZS accounting model is lower, *ceteris paribus*.

Therefore, Hypothesis 2 is supported; a double-entry automated accounting system is less risky to audit than a single-entry automated accounting system. QED

6 Discussion and conclusions

The mathematical elegance of EZS is well-documented, richly evidenced, and has stood the test of time [e.g., Mattessich, (1995), pp.5–40]. Despite centuries of business changes and the sophisticated creativity of business men and women, the ability of this system to encode every transaction into an equational zero vector in one of five basic account primitives has not been compromised. Mattessich (1995, pp.15–40) outlines the evolution of thought that went into which of those five accounts should be on the left side of the equation and which on the right to ensure the additive nature of the zero vectors; this in a time before calculators, when being able to add numbers, and not subtract them, was also a control device. Adding numbers leads to fewer errors than subtracting *and* adding numbers.

Historically, the double-entry framework was introduced to replace a single-entry framework in order to introduce the equational zero-account control lacking in a single-entry system, however sophisticated. It took centuries for this concept to gain full acceptance, but by the turn of the 20th century, double-entry accounting had been fully accepted as a necessary control. Today, this control feature of the double-entry accounting framework is as relevant as it was when it was first introduced. Indeed, in a post-SOX environment, this control feature may be more relevant than ever. Its relevance in a computerised environment is valid as well. Computers are designed to perform repetitive mathematical operations. Consequently, the ability to control data using equational zero vectors is a trivial operation in a computerised environment. Accounts either zero out or they do not. Attempting to build in a similar control in a NES would be extraordinarily difficult without transforming the NES into an EZS.

EZS has stood the test of time because it was designed as a control device. This control device was modified when it migrated from a strictly manual system to file-based systems; and modified yet again when it migrated to relational database systems and then evolved into object oriented systems. Yet, despite the loss of some of the control features of the EZS in some of these systems, the retention of the basic mathematic semantic with its built-in control feature of the equational zero vector account persists, as do the control accounts and the subsidiary accounts. They persist because, as others have said before us: “One thing expert designers know not to do is solve every problem from first principles. Rather, they reuse solutions that have worked for them in the past. When they find a good solution, they use it again and again. Such experience is part of what makes them experts.” [Gamma et al., (1995), p.1; Gerard, (2005), p.57]. The resilience of EZS over many centuries attests to this.

Modern NES introduce sophistication, parsimony and modelling of non-financial transactions into accounting system frameworks. Proponents of these systems are to be applauded for introducing and reinforcing the concept that non-financial information is important to capture in any accounting framework. But to the extent that these proponents make claims that the double-entry accounting framework is sub-standard, they malign

this model unfairly. EZS has stood the test of time because, as an ontology, it provides everything that a NES framework provides and also adds an element of control, *by design*, indeed, by its very nature, that a NES cannot match.

EZS is the application that financial accountants and auditors expect to find in AIS. Database issues aside, in order for NES to replace EZS as an accounting application, NES proponents would have to address auditor issues of internal control. The fact that financial accountants and auditors in practice have virtually ignored modern NES as a replacement application for EZS over the last quarter century speaks to the lack of acceptance of these models as a financial application. Just because one can build a NES accounting system does not mean one should migrate to a NES accounting system.

EZS is the generally accepted application to be modelled. REA, one of the more popular modern NES, has been shown to be a superior database design tool. The EZS is difficult conceptually because of the translation from an economic semantic into GAAP as well as a mathematic semantic. REA is an economic semantic. REA can be used as an aid in the translation from the economic semantic into GAAP. REA can also be used to design EZS databases. But for MIS and computer science, UML is the generally accepted modelling standard. Those using the REA model should be prepared to defend the use of REA, not against the EZS application model which is a 'strawman' exercise, but against the alternative UML modelling standard. It is in this direction that future research lies.

Acknowledgements

The authors thank the anonymous reviewers for their time and their thoughtful, constructive review comments, which have substantially improved the paper. We also thank participants for their input at of the Seventeenth Annual Research Workshop on Artificial Intelligence and Emerging Technologies in Accounting, Auditing and Tax, held in Anaheim, California, USA, on August 2, 2008.

References

- Alali, F., Grant, G.H. and Miller, K.C. (2008) 'IT control deficiencies that impact financial reporting', *Internal Auditing*, Vol. 23, No. 4, pp.28–38.
- Amir, E. and Lev, B. (1996) 'Value-relevance of non-financial information: the wireless communications industry', *Journal of Accounting and Economics*, Vol. 22, pp.3–30.
- Ashton, D., Dunmore, P. and Tippett, M. (2004) 'Double entry bookkeeping and the distributional properties of a firm's financial ratios', *Journal of Business Finance and Accounting*, Vol. 31, Nos. 5–6, pp.583–606.
- Ashworth, H.M. (1886) 'Some safeguards against fraud', *The Accountants' Students' Journal*, Vol. 3, No. 37, pp.37–41.
- Bardsley, J.B. (1900) 'Fraud in cash accounts: protection against and detection of', *The Accountants' Students' Journal*, September, pp.103–108.
- Barra, R. and Griggs, K. (2007) 'Internal controls: lessons to be learned from fire', *International Journal of Issues and Standards*, Vol. 3, No. 4, pp.375–389.
- Batra, D. and Sin, T. (2008) 'The READY model: patterns of dynamic behavior in REA-based accounting applications', *Information Systems Management*, Vol. 25, pp.200–210
- Bodnar, G. (1975) 'Reliability modeling of internal control systems', *The Accounting Review*, October, pp.747–757.

- Borthick, A.F. (2008) 'Auditing system development: constructing the meaning of 'systematic and rational' in the context of legacy code migration for vendor incentives', *Journal of Information Systems*, Vol. 22, No. 1, pp.47–62.
- Bradford, M., Richtermeyer, S. and Roberts, D. (2007) 'System diagramming techniques: an analysis of methods used in accounting education and practice', *Journal of Information Systems*, Vol. 21, No. 1, pp.173–212.
- Braun, D. (2001) 'Assets and liabilities are the momentum of particles and antiparticles displayed in Feynman-graphs', *Physica A*, Vol. 290, pp.491–500.
- Callaghan, J., Savage, A. and Peacock, E. (2002) 'Teaching REA semantics within an information engineering framework', *SA Journal of Accounting Research*, Vol. 16, No. 1, pp.59–80.
- Church, K.S. and Smith, R.E. (2007) 'An extension of the REA framework to support balanced scorecard information requirements', *Journal of Information Systems*, Vol. 21, No. 1, pp.1–25.
- COBIT 4.1 *Executive Summary Framework* (2007) IT Governance Institute, Rolling Meadows, Illinois, USA.
- Committee of Sponsoring Organizations (1992) *Internal Control – Integrated Framework*, AICPA, Jersey City, New Jersey, USA.
- Cooley, J. and Cooley, B. (1982) 'Internal accounting control systems: a simulation program for assessing their reliabilities', *Simulation & Games*, Vol. 13, No. 2, pp.211–231.
- Craig, A.T. (1898) 'Frauds in connection with bookkeeping, and methods to be used for their detection', *The Accountants' Journal*, Vol. 15, March, pp.295–300.
- Cushing, B.E. (1974) 'A mathematical approach to the analysis and design of internal control systems', *The Accounting Review*, January, pp.24–41.
- Dominguez, E., Rubio, A. and Zapata, M. (2002) 'Dynamic semantics of UML state machines: a metamodeling perspective', *Journal of Database Management*, Vol. 13, pp.20–38.
- Dunn, C.L. (2008) *NSF Workshop Website*, available at <http://www.msu.edu/user/mccarth4/NSFontology/positions.ppt>, accessed on 10 October 2008.
- Dunn, C.L. and Grabski, S.V. (2000) 'Perceived semantic expressiveness of accounting systems and task accuracy effects', *International Journal of Accounting Information Systems*, Vol. 1, pp.79–87.
- Dunn, C.L. and McCarthy, W.E. (1997) 'The REA accounting model: intellectual heritage and prospects for progress', *Journal of Information Systems*, Vol. 11, pp.31–51.
- Dunn, C.L., Gerard, G.J. and Grabski, S.V. (2005) 'Critical evaluation of conceptual data models', *International Journal of Accounting Information Systems*, Vol. 6, pp.83–106.
- Ellerman, D. (1982) *Economics, Accounting, and Property Theory*, D.C. Heath, Lexington, Massachusetts, USA.
- Ellerman, D. (1985) 'The mathematics of double entry bookkeeping', *Mathematics Magazine*, Vol. 58, September, pp.226–233.
- Ellerman, D. (1986) 'Double entry multidimensional accounting', *Omega*, Vol. 14, January, pp.13–22.
- Ellerman, D. (forthcoming) 'Mathematical formulation and generalization of double-entry bookkeeping', *FSR Forum*, Working paper available at <http://www.ellerman.org/Davids-Stuff/Maths/Math.htm#Gen-DEB>, accessed on 11 May 2008.
- Fields, K.T., Sami, H. and Sumners, G.E. (1986) 'Quantification of the auditor's evaluation of internal control in data base systems', *Journal of Information Systems*, Vol. 1, No. 1, pp.24–47.
- Fischer, R. and Braun, D. (2003) 'Nontrivial bookkeeping: a mechanical perspective', *Physica A*, Vol. 324, pp.266–271.
- Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Reading, MA.
- Geerts, G.L. and McCarthy, W.E. (2002) 'An ontological analysis of the economic primitives of the extended-REA enterprise information architecture', *International Journal of Accounting Information Systems*, Vol. 3, pp.1–16.

- Gerard, G.J. (2005) 'The REA pattern, knowledge structures, and conceptual modeling performance', *Journal of Information Systems*, Vol. 19, No. 2, pp.57–77.
- Grimlund, R.A. (1982) 'An integration of internal control system and account balance evidence', *Journal of Accounting Research*, Vol. 20, No. 2, pp.316–342.
- Hammer, M. and McLeod, D. (1981) 'Database description with SDM: a semantic database model', *ACM Transactions on Database Systems*, Vol. 6, No. 3, pp.351–386.
- Hessellund, A. (2006) 'Supply chain modeling with REA: IT University of Copenhagen, TR-2006-80', *University Technical Report Series*, pp.1–15.
- Ho, L.T., Lin, G. and Nagalingam, S. (2009) 'A risk mitigation framework for integrated-enterprise systems implementation for the manufacturing environment', *International Journal of Business Information Systems*, Vol. 4, No. 3, pp.290–310.
- International Organization for Standardization (2007) *Information Technology-Business Operational View – Part 4: Business Transaction Scenarios – Accounting and Economic Ontology*, International Organization for Standardization, Geneva, Switzerland.
- IT Governance Institute (2006) *IT Control Objectives for Sarbanes-Oxley: The Role of IT in the Design and Implementation of Internal Control over Financial Reporting*, 2nd ed., IT Governance Institute, Rolling Meadows, Illinois, USA.
- Javidi, M.M, Naji, R.H. and Shiri, M.E. (2009) 'Increasing the security of e-commerce systems using multi-agents', *International Journal of Business Information Systems*, Vol. 4, No. 4, pp.460–476.
- Jones, R.A. (2000) 'Sizing up NPO software', *Journal of Accountancy*, November, pp.28–44.
- Jones, R.A. (2002) 'Spotlight on midlevel ERP software', *Journal of Accountancy*, May, pp.24–47.
- Jones, R.A. (2007) 'NPO software measures up', *Journal of Accountancy*, July, pp.68–72.
- Jones, R.A., Tsay, J. and Griggs, K. (2002) 'An empirical investigation of the cognitive fit of selected process model diagramming techniques', *Review of Business Information Systems*, Vol. 6, No. 4, pp.101–113.
- Jones, R.A., Tsay, J. and Griggs, K. (2005–2006) 'An empirical investigation of the task specific relative strengths of selected accounting and information systems diagramming techniques', *Journal of Computer Information Systems*, Vol. 46, No. 2, pp.99–114.
- Joosten, S. and Purao, S. (2002) 'A rigorous approach for mapping workflows to object-oriented IS models', *Journal of Database Management*, Vol. 13, pp.1–19.
- Kim, H.J. (2009) 'New accounting system for the QoS-enabled multiservice internet: technical overview', *International Journal of Business Information Systems*, Vol. 4, No. 2, pp.173–194.
- Klamm, B.K. and Weidenmier, M.L. (2004) 'Linking business processes and transactions cycles', *Journal of Information Systems*, Vol. 18, No. 2, pp.113–125.
- Knechel, W.R. (1983) 'The use of quantitative models in the review and evaluation of internal control: a survey and review', *Journal of Accounting Literature*, Vol. 2, pp.205–219.
- Kumar, V., Pollanen, R. and Maheshwari, B. (2008) 'Challenges in enhancing enterprise resource planning systems for compliance with Sarbanes-Oxley Act and analogous Canadian legislation', *Management Research News*, Vol. 31, No. 10, pp.758–773.
- Larsen, T., Niederman, F., Limayem, M. and Chan, J. (2007) 'The role of modelling in achieving information systems success: UML to the rescue?', *Information Systems Journal*, Vol. 19, pp.83–117.
- Lundholm, R.J. (1995) 'A tutorial on the Ohlson and Feltham/Ohlson models: answers to some frequently asked questions', *Contemporary Accounting Research*, Vol. 11, No. 2, pp.749–761.
- Marshall, H.C. (1895) 'Questions that arise in practice', *The Accountants' Journal*, Vol. 13, August, pp.105–111.
- Mattessich, R. (1995) *Critique of Accounting: Examination of the Foundations and Normative Structure of an Applied Discipline*, Quorum Books, Westport, Connecticut, USA.
- McCarthy, W.E. (1982) 'The REA accounting model: a generalized framework for accounting systems in a shared data environment', *The Accounting Review*, July, pp.554–578.

- McCarthy, W.E. (2003) 'The REA modeling approach to teaching accounting information systems', *Issues in Accounting Education*, Vol. 18, No. 4, pp.427–441.
- National Science Foundation (2007) 'Ontological specification of interoperability semantics for financial information systems', 2 April, (W. McCarthy), available at <http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0749219>, accessed on 10 October 2008.
- O'Leary, D. (2004) 'On the relationship between REA and SAP', *International Journal of Accounting Information Systems*, Vol. 5, pp.65–81.
- Poels, G., Maes, A., Gailly, F. and Paemeleire, R. (2007) 'The pragmatic quality of resources-events-agents diagrams: an experimental evaluation', *Information Systems Journal*, pp.1–27.
- Rockwell, S.R. and McCarthy, W.E. (1999) 'REACH: automated database design integrating first-order theories, reconstructive expertise, and implementation heuristics for accounting information systems', *International Journal of Intelligent Systems in Accounting, Finance and Management*, Vol. 8, pp.181–197.
- Romney, M.B. and Steinbart, P.J. (2009) *Accounting Information Systems*, 11th ed., Prentice-Hall, Upper Saddle River, New Jersey, USA.
- Sarbanes-Oxley Act of 2002*, 107 Congress of the United States of America, Second Session.
- Sedbrook, T. and Newmark, R.I. (2008) 'Automating REA policy level specifications with semantic web technologies', *Journal of Information Systems*, Vol. 22, No. 2, pp.249–277.
- Selic, B. (2006) 'UML 2: a model-driven development tool', *IBM Systems Journal*, Vol. 45, No. 3, pp.607–620.
- Solari, C. (2007) 'Designing for security', *Bell Labs Technical Journal*, Vol. 12, No. 3, pp.1–6.
- Srinidhi, B.N. (1988) 'Mathematical formulation of the task segregation problem in internal control system design', *Decision Sciences*, Vol. 19, pp.1–16.
- Sutton, S.G. and Arnold, V. (2005) 'The Sarbanes-Oxley Act and the changing role of the CIO and IT function', *International Journal of Business Information Systems*, Vol. 1, Nos. 1–2, pp.118–128.
- Verdaasdonk, P. (2003) 'An object-oriented model for ex ante accounting information', *Journal of Information Systems*, Vol. 17, No. 1, pp.43–61.
- Vymetal, D., Hucka, M., Hunka, F. and Kasik, J. (2008) 'Production planning model using REA ontology', *Ekonomie A Management*, Vol. 4, pp.93–102.
- Weber, R. (1980) 'Some characteristics of the free recall of computer controls by EDP auditors', *Journal of Accounting Research*, Vol. 18, No. 1, pp.214–241.
- Yu, S. and Neter, J. (1973) 'A stochastic model of the internal control system', *Journal of Accounting Research*, Vol. 11, pp.273–295.

Notes

- 1 Consider a complex exchange transaction in which a fixed asset is exchanged for another fixed asset and cash is received or possibly given up. Three resources are involved in this transaction: old fixed asset, new fixed asset and cash. In an EZS, one journal entry that balances would be generated to record this transaction. In a NES, three entries, at a minimum, would need to be generated: one to show the disposal of the old fixed asset, one to show the new fixed asset, and one to show the receipt or disbursement of cash. This provides three opportunities to make mistakes/errors. While the three entries are clearly related, the database designer has to build into the system controls to tie these three transactions together, but controls also need to be included to ensure that the cash is recorded in the correct direction and that the fixed assets will be correctly added or subtracted from the fixed asset resource table. Such controls are inherent in the EZS.

Appendix A Mathematical formulation and generalisation of double-entry bookkeeping

(Quoted from Ellerman, forthcoming)

“The double-entry method of accounting is a method of using the Pacioli group to perform additive algebraic operations on equations such as the conventional balance sheet equation:

$$\text{Assets} = \text{Liabilities} + \text{Net Worth.}$$

Balance-sheet equation

Vector equations are first encoded in the Pacioli group. A T-account equal to the zero T-account $[0 // 0]$ is called a zero-account. Equations encode as zero-accounts. Since the vectors in a T-account must be non-negative, we must first develop a way to separate out the positive and negative components of a vector. The positive part of a vector x is $x^+ = \max(x, 0)$, the maximum of x and the zero vector [note that ‘0’ is used depending on the context to refer to the zero scalar or the zero vector]. The negative part of x is $x^- = -\min(x, 0)$, the negative of the minimum of x and the zero vector. Both the positive and negative parts of a vector x are non-negative vectors. Every vector x has a ‘Jordan decomposition’ $x = x^+ - x^-$.

Given any equation in \mathbf{R}^n , $w + \dots + x = y + \dots + z$, each left-hand side (LHS) vector w is encoded as a debit-balance T-account $[w^+ // w^-]$ and each right-hand side (RHS) vector y is encoded as a credit-balance T-account $[y^- // y^+]$. Then the original equation holds if and only if the sum of the encoded T-accounts is a zero-account:

$$w + \dots + x = y + \dots + z$$

if and only if

$$[w^+ // w^-] + \dots + [x^+ // x^-] + [y^- // y^+] + \dots + [z^- // z^+]$$

is a zero-account.

Encoding an equation as a zero T-account

Given the equation, the sum of the encoded T-accounts is called the equational zero-account of the equation. Since only plus signs can appear between the T-accounts in an equational zero-account, the plus signs can be left implicit. The listing of the T-accounts in an equational zero-account (without the plus signs) is the ledger.

Changes in the various terms or ‘accounts’ in the beginning equation are recorded as transactions. Transactions must be recorded as valid algebraic operations which transform equations into equations. Since equations encode as zero-accounts, a valid algebraic operation would transform zero-accounts into zero-accounts. There is only one such operation in the Pacioli group: add on a zero-account. Zero plus zero equals zero. The zero-accounts representing transactions are called transactional zero-accounts. The listing of the transactional zero-accounts is the journal.

A series of valid additive operations on an equation can then be presented in the following standard scheme:

$$\begin{aligned} & \text{Beginning Equational Zero-Account} \\ & + \text{Transactional Zero-Accounts} \\ & = \text{Ending Equational Zero-Account} \end{aligned}$$

or, in more conventional terminology,

$$\begin{aligned} & \text{Beginning Ledger} \\ & + \text{Journal} \\ & = \text{Ending Ledger} \end{aligned}$$

The process of adding the transactional zero-accounts to the initial ledger to obtain the ledger at the end of the accounting period is called posting the journal to the ledger. The fact that a transactional zero-account is equal to $[0 // 0]$ is traditionally expressed as the double-entry principle that transactions are recorded with equal debits and credits. The summing of the debit and credit sides of a computed equational zero-account to check that it is indeed a zero-account is traditionally called the trial balance.

It remains to decode the ending equational zero-account to obtain the equation that results from the algebraic operations represented in the transactions. The T-accounts in an equational zero-account can be arbitrarily partitioned into two sets debit balance (DB) and credit balance (CB). T-accounts $[w // x]$ in DB are decoded as $w - x$ on the left side of the equation, and T-accounts $[w // x]$ in CB are decoded as $x - w$ on the right side of the equation. Given a zero-account, this procedure yields an equation. In an accounting application, the T-accounts in the final equational zero-account would be partitioned into sets DB and CB according to the side of the initial equation from which they were encoded.”

Appendix B A simple example of multidimensional property accounting

When the scalars (single non-negative numbers) of value accounting are replaced by non-negative vectors, then the vectors can be interpreted as representing the physical amounts of different types of property. Consider a simple model where there are only three types of property: cash, outputs and inputs. These goods will be listed in that order in each three-dimensional vector.

Let the initial asset vector be $(900, 20, 30)$ so the firm has \$900 cash, 20 units of the output in inventory, and 30 units of the input in inventory. The firm also has a \$400 liability represented by the vector $(400, 0, 0)$ so the equity vector (Assets – Liabilities) is given by the vector $(500, 20, 30)$. Thus the initial balance sheet (vector) equation is:

$$\begin{array}{lclcl} \textit{Assets} & = & \textit{Liabilities} & + & \textit{Equity} \\ (900, 20, 30) & = & (400, 0, 0) & + & (500, 20, 30). \end{array}$$

Initial vector balance-sheet equation

This encoded as the following equational zero-account or ledger:

$$\begin{array}{lcl} \textit{Assets} & & \textit{Liabilities} & & \textit{Equity} \\ [(900, 20, 30)//(0, 0, 0)] & & [(0, 0, 0)//(400, 0, 0)] & & [(0, 0, 0)//(500, 20, 30)]. \end{array}$$

Initial vector T-accounts in ledger

Consider the following transactions:

- 1 10 units of the product are produced
- 2 5 units of the product are sold for \$50 each
- 3 \$200 principal payment is made on a loan.

These transactions are then encoded as transactional zero-accounts and added to the ledger T-accounts:

	<i>Assets</i>	<i>Liabilities</i>	<i>Equity</i>
	[(900, 20, 30)/(0, 0, 0)]	[(0, 0, 0)/(400, 0, 0)]	[(0, 0, 0)/(500, 20, 30)].
1	[(0, 10, 0)/(0, 0, 0)]		[(0, 0, 0)/(0, 10, 0)]
2	[(250, 0, 0)/(0, 5, 0)]		[(0, 5, 0)/(250, 0, 0)]
3	[(0, 0, 0)/(200, 0, 0)]	[(200, 0, 0)/(0, 0, 0)]	
	[(1150, 30, 30)/(200, 5, 0)]	[(200, 0, 0)/(400, 0, 0)]	[(0, 5, 0)/(750, 30, 30)]
=	[(950, 25, 30)/(0, 0, 0)]	[(0, 0, 0)/(200, 0, 0)]	[(0, 0, 0)/(750, 25, 30)]

Initial Vector Ledger + Journal = Ending Vector Ledger

where the last line of ledger accounts is in reduced form. The reduced accounts are then decoded to obtain the ending balance-sheet equation:

<i>Assets</i>	<i>Liabilities</i>	<i>Equity</i>
(950, 25, 30)	(200, 0, 0)	(750, 25, 30)

Ending Vector Balance-Sheet Equation

Many software vendors already encode non-financial information as part of the journalising process, using various user interfaces to accomplish the encoding.