Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2007 Proceedings

European Conference on Information Systems (ECIS)

2007

Consideration of Risks and Internal Controls in Business Process Modeling

R. Mansour
University of South Florida, rmansour@coba.usf.edu

U. Murthy
University of South Florida, umurthy@cob.usf.edu

CONSIDERATION OF RISKS AND INTERNAL CONTROLS IN BUSINESS PROCESS MODELLING

Mansour, Rosalyn, University of South Florida, 4202 E. Fowler Ave. BSN3403, Tampa, FL 33620-5500, rmansour@coba.usf.edu

Murthy, Uday S., University of South Florida, 4202 E. Fowler Ave. BSN3403, Tampa, FL 33620-5500, umurthy@coba.usf.edu

Abstract

Given the myriad risks facing organizations these days, information systems and more importantly the data underlying the systems are susceptible to material errors, irregularities, or even fraud. It is therefore critically important to ensure that proper controls are built into organizational information systems. This paper describes an internal control oriented ontological extension to McCarthy's (1982) Resources, Events, Agents model. The ontological extension facilitates the identification and documentation of internal controls at the business process level. With a focus on accounting information systems, we first show how the basic REA framework is used to model revenue cycle business processes. We then identify illustrative risks, the corresponding audit objectives, and related internal control procedures for the sales order processing subsystem. A UML diagram of the sales order processing subsystem entities is shown, with specific table and field level controls indicated. Future directions in this line of research aimed at refinement of the internal control ontology is also discussed.

Keywords: Accounting Information Systems, Resources Events Agents Framework, Internal Control, Ontology.

1 INTRODUCTION

In the United States, the Sarbanes-Oxley Act of 2002 imposes requirements on company management to make annual assessments of its internal controls and on auditors to attest to the state of the company's internal control system. The requirements of the Sarbanes-Oxley Act impacted all companies, including European companies with securities traded in the United States. Worldwide, companies are under increased pressure not only to ensure that internal controls are designed into their corporate information systems but also to facilitate their ongoing review and evaluation. Among other things, compliance with Sarbanes-Oxley requires adequate documentation of the internal controls in the company's information systems. This documentation of internal controls can be quite complex to create and even more tedious for internal and external auditors to analyze for assessing whether the internal controls are effective. The problem of documentation of information systems controls is further exacerbated by the fact that business processes, and consequently internal controls, vary considerably across industries. Furthermore, existing conceptual data modelling approaches such as entity-relationship (ER) diagrams and UML diagrams do not contain modelling primitives specifically for defining internal controls. Systems analysts, designers, and auditors could all benefit from enhancements to existing conceptual modelling techniques that would facilitate the modelling of internal controls in the system.

Conceptual modelling approaches employing ER and UML diagrams often have an ontological purpose, since the objective is to communicate the semantics of the domain to different users and enable information interchange. Ontologies are simply agreed-upon definitions that articulate what a phenomenon is and consist of "…formal descriptions of entities and their properties, relationships, constraints, and behaviors" (Fox et al 1996, p.124). Ontologies also provide theory upon which models, or representations, of phenomena can be grounded (Weber, 2002). Such representations can then be used to define, communicate, refine, and improve the real-world phenomenon of interest. Ultimately, the representations are implemented in some form in functioning information systems. One ontology geared specifically toward modelling business processes is McCarthy's (1982) Resources, Events, and Agents (REA) framework. The REA framework is focused on representing information about economic resources, economic events, and economic agents and relationships among them. In an enterprise-wide information processing context, using the REA framework as the basis for constructing an ER diagram results in "event entities" such as sales and purchases, "resource entities" such as cash and inventory, and "agent entities" such as customers and suppliers. While the REA model has been shown to be robust for modelling a variety of accounting and business phenomena, it lacks ontological features specifically for modelling the internal control features of an information system.

The purpose of this paper is to describe an internal control oriented ontological extension to McCarthy's REA model. According to Weber (2002), the REA model is a "material" ontology that models organizational phenomena with the *de facto* goal of eventual implementation in a database. Organizational phenomena undergo changes via business processes. Unless proper internal controls are in place, the changes caused can result in material errors, irregularities, or even fraud. Thus, it is critically important to ensure that controls are built into organizational information systems. Accordingly, the goal of this paper is to describe an ontological extension to the REA model to facilitate the identification and documentation of internal controls at the business process level. The proposed internal control ontology can be instantiated using either a relational database representation or an object-oriented representation of a business information system using UML. With the increasing object-orientation of business information systems, the proposed ontology fits in well with Geerts and McCarthy's (2002) call for the REA approach to be further integrated with the object-oriented paradigm and particularly with UML.

2 EXISTING CONTROL ONTOLOGIES

There has been some focus in the information systems literature on the topic of developing ontologies for various purposes. According to Weber (2002), ontologies can be categorized based on their level of generality, granularity, and structure. A "material" ontology, as defined by Weber (2002, p.14), is one that focuses on "sub-regions of reality." The internal control ontology proposed in this paper would be classified as a material ontology.

Wand and Weber (1989) describe an ontology of an accounting information system at a very general level of abstraction. Although they use a payroll system for illustration purposes, the Wand and Weber (1989) ontology is at a sufficiently high level of abstraction to be applicable to any information system. As such, the concepts outlined in the Wand and Weber (1989) ontology can be easily incorporated into lower-level material control ontologies. Wand and Weber (1989) address the concept of control through the notion of whether the states and events in an information system are "lawful" or "unlawful." The applications that feed data to and manipulate data in the information system can cause either lawful or unlawful changes and it is the objective of controls to prevent unlawful changes. Importantly, the Wand and Weber (1989) control concept does not address controls at the database level, i.e., controls that govern unlawful conditions in a database regardless of the applications that feed the database.

The ontology proposed in the current paper is linked to the formal concepts of control contributed by Wand and Weber (1989). We extend the Wand and Weber definition of a "system" to include not only an application system but also the relational database that houses the application data. In our proposed ontology, most of the controls needed for an accounting information system will reside at the relational database level; therefore, only residual controls will exist at the application level.

An earlier attempt at developing an internal control model of a system was presented by Bailey et al. (1985). They developed a computerized analytical internal control modelling tool called "TICOM" (for The Internal Control Model). Auditors could use structured commands in "internal control design language" (ICDL) to query the TICOM model with the objective of evaluating the firm's internal control system. ICDL consists of the following object classes: agents (individuals, groups, divisions, or subsystems); objects (forms, records, assets, and other accounting system abstractions); repositories (storage for objects); and commands (actions upon objects). It is possible to model relationships between two classes. For example, a payroll check (type of class) is modelled as a check's (object class) relationship to the payroll department (agent class). Likewise, a payroll clerk would be modelled as a relationship between a clerk (agent class) and the payroll department (another agent class). Based on the Fox et al. (1996) definition, TICOM is an ontology of sorts; however, it has not gained much support in the systems design and development arena.

Kaplan et al. (1998) describe an approach to assessing data quality in accounting information systems (AIS) using a decision support systems approach. They implemented a set of mathematical models and algorithms that can be used to help an assessor (auditor) evaluate the data quality of an AIS or to design testing procedures to ensure that an AIS meets specific data quality goals. While there is clearly a connection between data quality and internal controls, the Kaplan et al. modelling approach is *ex post* oriented, seeking to assess the data quality of an existing system. The approach we propose in this paper is *ex ante* oriented, seeking to identify and document internal controls at the systems analysis and design stage.

The only prior research in REA that is somewhat related to the goals of this paper is the work by Geerts and McCarthy (2002) who refer to type-level specification extensions to the REA model. Type images can be thought of as generalizations of an entity type that are aimed at grouping similar categories of the entity. Type-level specifications can be used to categorise entities into control categories, as in the example provided by Geerts and McCarthy (2002) where customers are typed as either "high credit risk" or "low credit risk" customers. The

approach proposed in the current paper is broader and seeks to encompass a wide variety of internal controls into the conceptual data model.

3 BUSINESS PROCESS MODELING USING THE REA FRAMEWORK

Since traditional accounting models of debits and credits and other types of artifacts do not harness the power of present day information systems that use relationship database structures. McCarthy (1979, 1982) sought to "reengineer" the accounting model through the "Resources, Events, and Agents" model that steered clear of accounting artifacts like debits and credits. Using the REA model, the developer models accounting phenomena by considering the three basic types of economic entities - resources, events, and agents. In effect, resources, events, and agents are the "building blocks" of economic phenomena. Resource-event-agent (REA) sets are related to one another by one of four types of relationships: stock-flow (connect resources to events and events to agents), duality (each increment in a resource caused by an REA set will have a corresponding decrement), control (connect event to both internal and external agents), and responsibility (connect inside agents to their superiors). Although the original REA model (McCarthy, 1979, 1982) has been extended to include such elements as time (O'Leary, 1999) and commitments (Geerts and McCarthy 2002), the structure of McCarthy's earlier models (1979, 1982) will be the frame of reference for the proposed study. Figure 1 shows an REA view for the sales and cash receipts processes within the revenue subsystem for a retailing enterprise.

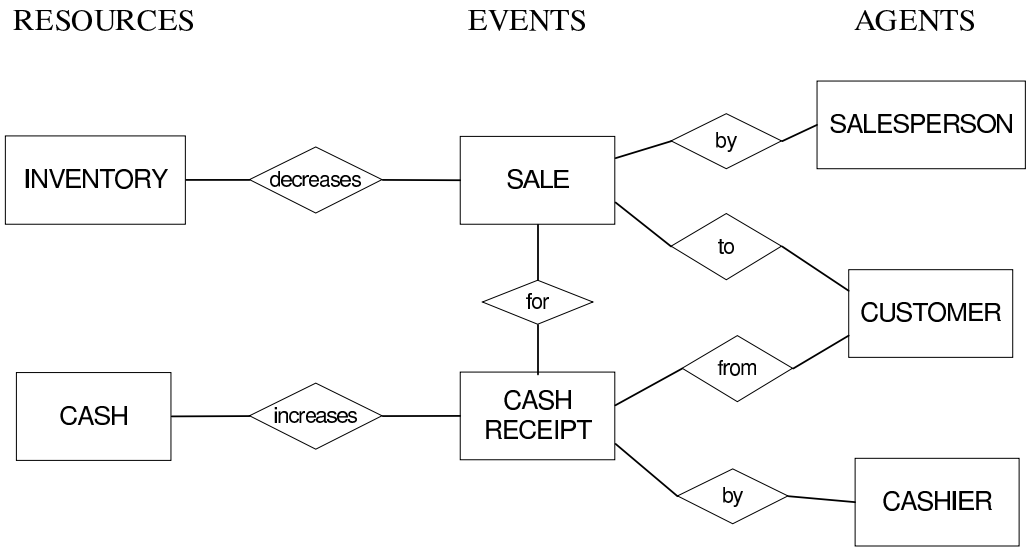

Figure 1: REA View for Credit Sales and Cash Receipts Processes

The REA model is well-suited for modelling real-world accounting/economic phenomena. It seeks to capture information about the economic exchanges (events), the resources that are affected by the exchanges, and the agents (internal and external) who are involved. In Figure 1, the resources (inventory, cash) are linked to the events (sale, cash receipt) in what are referred to in REA terminology as "stock-flow" relationships, because the events cause outflows (inventory outflow) or inflows (cash inflow) of resources. The relationships between events and agents (by, to, from) are referred to in REA terminology as "control" relationships. Finally, relationships between events are referred to as "duality" relationships, which pair resource increasing (decreasing) events with corresponding resource decreasing (increasing) events. In Figure 1, the "sale" event is a resource decreasing event that is paired with "cash receipt" which is a resource increasing event.

In the REA framework, control relationships explicitly focus on internal controls in terms of the internal agent responsible for performing an event. Beyond control relationships, however, the REA modelling framework does not focus explicitly on other event- or resource-related internal controls that are or should be present within the domain. Accordingly, the REA model lacks modelling primitives for depicting such internal controls. As indicated earlier, corporate governance requirements imposed on companies have heightened the need for internal controls in their information systems. Since it is more expensive and difficult to modify existing systems to embed controls, the early consideration of internal controls in the systems analysis and design process would facilitate their incorporation into systems at lower cost. Accordingly, our objective is to embellish the REA framework with internal control modelling primitives so that systems analysts and designers can consider internal control (including access controls for security purposes) at an early stage in the systems development life-cycle.

3.1 Audit Objectives

In the process of auditing the financial statements of a company, auditors seek to achieve a set of audit objectives. At the financial statement account level, the primary audit objectives are existence/occurrence, completeness, accuracy, classification, and timing, and presentation and disclosure (Arens and Loebbecke, 1997). Of these objectives, "presentation and disclosure" relates to the appearance of accounts on the company's published financial statements and as such does not apply at the transaction level.

The *existence* (or occurrence) objective seeks to determine whether an event that has been captured in the system actually occurred. For example, a sales order from a fictitious customer is not a true and valid sale. A corollary to the existence objective is the *completeness* objective, which seeks to determine whether all events that occurred have been recorded in the information system. This objective is more difficult to verify in comparison with the existence objective, since the auditor would be looking for evidence of events having occurred outside of the boundaries of the information system.

Accuracy implies that an event is recorded at the correct amount and that the amount recorded accurately reflects the event that occurred. As the term implies, the accuracy objective is aimed at ensuring that data pertaining to different aspects of the event, such as the date, quantities, prices, and transaction codes are all properly applied. The *classification* objective is concerned with other measures of exactness in the representation of the event. A classification problem would arise if, for example, an incorrect agent or resource were attached to an event such as when a sale is identified with the wrong customer. The *timing* objective seeks to ensure that events are recorded on the date the transaction took place. Violations of the timing objective would result in the financial statements not accurately presenting the results of transactions for the reporting period to which the financial statements relate.

3.2 Risks in Transaction Processing Systems

For each transaction processing information system, it is necessary to consider the risks that are present, the audit objectives relating to those risks, and the control procedures that are or should be in place to mitigate the risks. Formally, "risk" in the context of information systems is defined as the vulnerabilities, threats, and the susceptibility to error of an organization's information assets. This definition of risk is narrower than that in international standards such as ISO 27001 and ISO/IEC 17799:2005 (ISO/IEC 2005a, 2005b). Our concept of risk is restricted to the threats to the accuracy of reported information, whereas the definition of risk in these ISO standards is much broader and includes aspects such as confidentiality and availability of information.

An important step during systems analysis and design is to consider the risks to accurate information that are present in the application domain. It is critical for the systems analyst and

auditor to work together to identify these risks. The auditor will likely consider and identify these risks from an audit objective and accounting and business process perspective. The systems analyst will likely consider these risks from an information systems perspective and from the standpoint of the features of the specific information technology environment. Both perspectives are necessary to identify the maximum number of risks to be guarded against, ensure that all audit objectives are being satisfied, and increase the possibility that the control capabilities of the information system are put to full use. It is our contention that relating risks and controls to audit objectives leads to a more comprehensive consideration of risks and hence controls. Additionally, relating risks and controls to audit objectives provides common language between systems analysts and auditors, thereby enhancing the process. Shown below in Table 1 are illustrative risks, related audit objectives, and controls that can be implemented for the sales order processing subsystem.

Risks	Audit objectives	Controls
Incorrect sales order date	Timing	Autodate
Invalid sales order identifier	Completeness	Autonumbering
Invalid inventory item on sales order	Classification	Check digit or closed loop verification
Invalid customer on sales order	Classification	Check digit, closed loop verification, referential integrity
Sales order placed by customer who is not an approved customer	Existence	Closed loop verification
Invalid salesperson on sales order	Classification	Check digit, closed loop verification, referential integrity
Invalid order quantity (quantity ordered is greater than inventory quantity on hand)	Accuracy	Reasonableness check or custom programmed control
Order placed by customer with insufficient or poor customer credit	Classification	Reasonableness check or approval
Inventory item ordered is unreasonable for customer (expensive item never ordered before)	Existence	Reasonableness check or custom programmed control
High amount sales order processed by unauthorized salesperson	Existence	Closed loop verification or programmed custom control
Incorrect calculations on sales order (quantity x price)	Accuracy	Calculated fields
Sales discounts not properly authorized and approved	Accuracy	Programmed custom control

Table 1 – Risks, audit objectives, and controls for sales order processing system

4 MODELLING CONTROL ACTIVITIES

In the context of this study, "internal control" is defined as a process that provides reasonable assurance that audit objectives are met and that the financial reporting process is reliable (COSO, 1992). Per the COSO framework, internal control objectives are achieved through an appropriate control environment, risk assessment, control activities, information and communication, and monitoring. Of interest is the "control activities" component of the COSO framework, which for the purpose of this paper is defined as *the policies and procedures that carry out the internal control objective of reliable financial reporting*. Control activities can be preventive, detective, or corrective, but preventive controls are preferred because they thwart errors from ever entering the accounting information system. Thus, while detective and corrective controls are important, this study's scope is preventive

control activities. Future research could consider extending our approach to detective and corrective controls.

We now discuss how preventive control activities can be modelled in the context of the audit objectives outlined in the previous section. Two modelling primitives are introduced: (1) a symbol for referencing a related REA diagram for separate depiction of control procedures (for resource and agent entities) and (2) representation of event-specific audit objectives (for event entities). The "V" with a number symbol is used to reference a separate REA view in which control procedures for the particular resource and agent entity are defined. Audit objectives relevant for each event are indicated by means of an box attached to the top of each event entity. Depicted in Figure 2 is an REA view of the sales order processing system with additional elements.

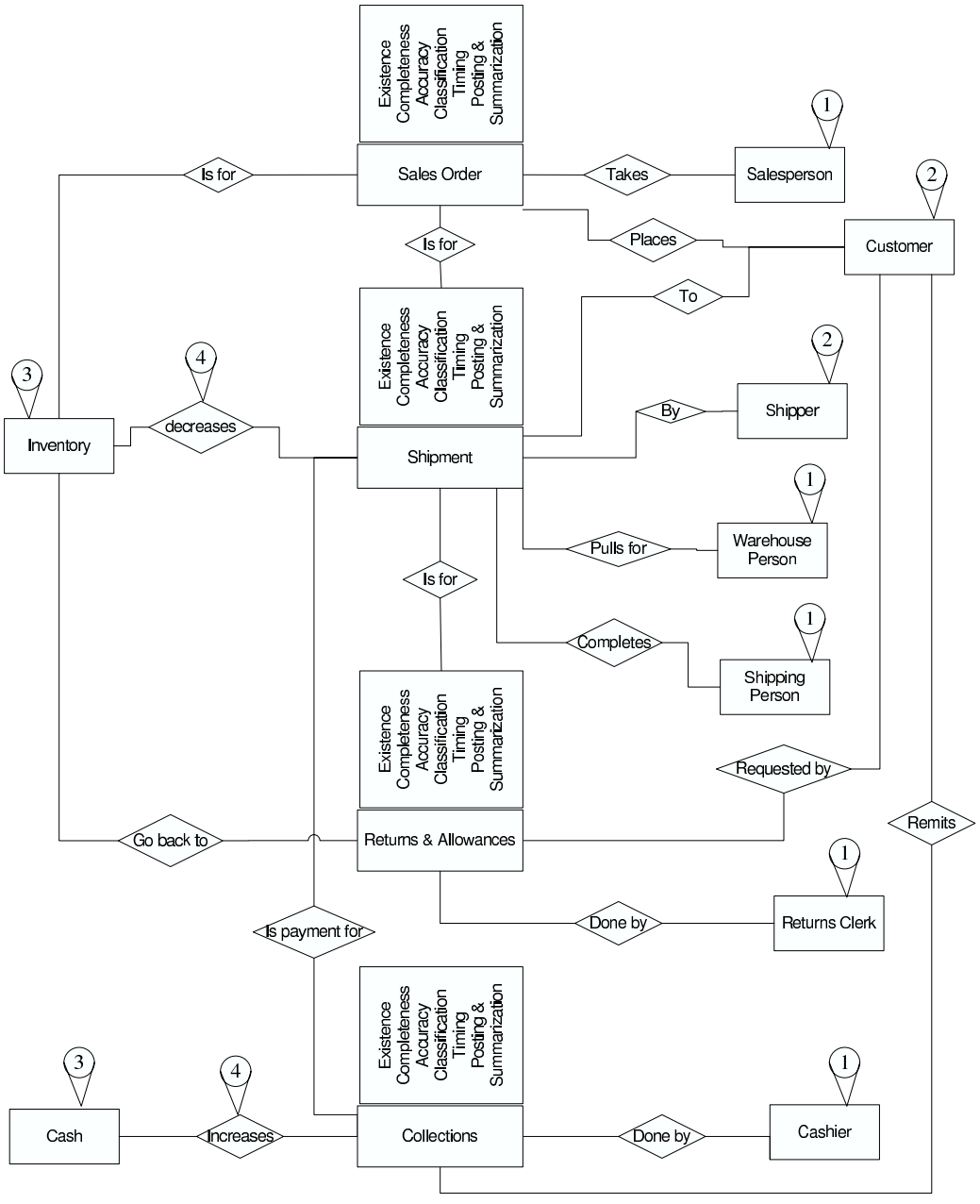

Figure 2: REA View of Sales Order Processing System with Reference Symbols and Audit Objectives

The first element to note in Figure 2 is the "V" with the number attached to agent entities, resource entities, and the relationships affecting resource entities (i.e., events causing resource increments and decrements). The number in the circle indicates the REA view that contains details of the processes, with corresponding controls, for the agent entity, resource entity, and resource changing events. For example, in Figure 2, details of the control procedures governing internal agents ("Salesperson," "Warehouse Person" etc.) are provided in REA view number 1. Details of the control procedure governing external agents ("Customer" and "Shipper") are provided in REA view number 2. Similarly, details of the control procedures governing resources ("Inventory" and "Cash") are provided in REA view number 3, while details of the control procedures governing resource increments and decrements are provided in REA view number 4. There is, of course, a high degree of interdependence between the REA views—the sales order processing view depends on the controls in the separate resource and agent views (referenced via the numbers on top of the "V" symbols) to ensure that all resources and agents accessed in the sales order processing subsystem are valid. Likewise, the resource and agent maintenance REA views depend on the controls in the sales order processing REA view to ensure that updates to them performed in the sales order processing view are valid.

In a business process REA diagram, the *event* entities are the main focus and consequently control procedures for event entities are of chief interest. Control procedures are not depicted for resource and agent entities on the main event view REA diagram. Rather, separate REA views are created for the maintenance processes necessary for resource and agent entities and the control procedures surrounding the maintenance processes are accordingly depicted on these separate REA views. Thus, on the business process REA diagrams, the reference symbols are used for resource and agent entities, to indicate the specific REA view in which control procedures for those entities can be found.

The second element added to Figure 2 is the add-on box attached to each event entity, containing the audit objectives relevant for that event. These are the transaction-related audit objectives to be fulfilled for the event to be considered "valid." The relationship between these audit objectives, risks, and control procedures was shown previously in Table 1.

4.1 Instantiation of Controls enhanced REA model of Revenue Cycle with UML Notation

Supporting the audit objectives and corresponding management assertions are specific control activities. A lower abstraction or instantiation of the revenue cycle using UML notation is shown in Figure 3 (next page). Only the sales order processing subsystem within the revenue cycle is shown. However, the remaining revenue cycle subsystems (shipment, collections, returns and allowances) would be modelled along similar lines. In all the figures, only attributes necessary for a valid event are shown for each resource and agent in the revenue cycle view. For example, commission rate would be an attribute of a salesperson, but since this is not of interest in the revenue cycle, this attribute is not shown for employee in Figure 3, whereas it would be shown as an attribute in the part of the payroll cycle (a different view) as would controls designed to prevent an invalid commission rate. For all entities in the REA sets, example data is provided to explain how the controls are to function.

To be used in conjunction with Figure 3, Table 1 shown earlier lists the risks specific to the sales order processing component of the revenue cycle and the related audit objectives. This list is intended to be illustrative and not comprehensive, but should be reasonably representative of what the risks in most real-world environments. The control procedures in Figure 3 are indicated at the table and field level and can be linked with the risks and audit objectives in Table 1. For example, one risk in Table 1 is that the sales order date is incorrect. As shown in Figure 3, the control to mitigate that risk is to make the "date" field in the sales order table an "autodate" field. As another example, another risk is that the customer number indicated on the sales order is invalid. As shown in Figure 3, the control that would mitigate

that risk is to establish referential integrity between the CNO (customer number) field in the sales order table and the CNO field in the customer table.

As alluded to earlier, while in the revenue cycle view one must assume that the Inventory entity is valid (all data is truly represented). Therefore, assuming the inventory price is valid, referential integrity will allow this valid inventory price to flow through to the sales order. This logic extends to all resource and agent entities. An entity-relationship diagram for the "customer" agent would show a set of controls that should be in place to specifically ensure that the customer is really a valid customer and that all additions, deletions, and modifications to the customer table are accurate, complete, and valid.

The controls in place for the sales order event are now described. First, assuming the customer, inventory, and salesperson entities are valid, referential integrity assures that these entities will exist in the sales order. Entity integrity in the sales order table prevents the sales order from being invalid because it prevents null values in the primary key (sales order number). In addition to these controls related to agents and events, the sales order event itself has a set of controls that not only assure its validity, but are relied upon by other REA sets in the same cycle, such as the shipment, collection, and returns and allowances subsystems. For example, the shipment event requires a valid sales order, which can be confirmed by referencing the sales order event table. Note also that every event in the revenue cycle has one set of controls in common, specifically automatic dating of transactions, automatic numbering (a sequence check could also be used), entity integrity, and access restrictions.

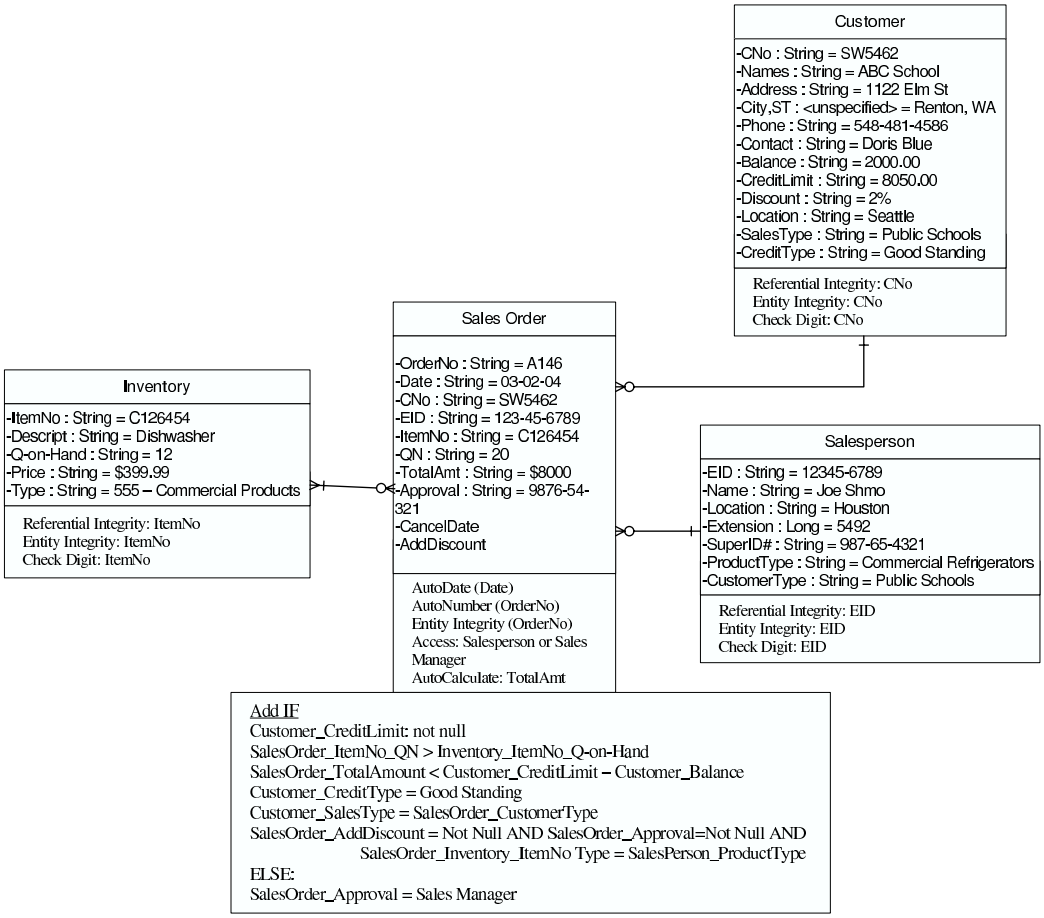

Figure 3: Controls to be represented in sales order processing subsystem – UML View

Automatic dating of a sales order ensures that the sales order date for a sales order event will always be valid (i.e., the current date in force when the sales order is recorded will automatically be applied as the sales order date). An accurate sales order date and shipping date is used to determine cut-off for sales shown in the financial statements (i.e., to ensure that only sales for the reporting period are included). The automatic numbering of sales orders seeks to ensure that all sales orders taken actually enter the accounting information system. This is not usually a concern because there is less motivation to underreport sales than to inflate sales; however, automatic numbering of the sales orders should alert someone in the organization if a customer's order is misplaced and not entered. Entity integrity ensures that duplicate sales orders cannot exist, again allowing events that occur later in the cycle to assume that the sales order is valid. Through the concept of referential integrity, it is possible to ensure that required preceding events have occurred before any given event is executed. For example, a business rule may specify that a credit check event must be completed before a sales order can be processed. This rule can be enforced by establishing referential integrity between the sales order and credit check events such that a sales order cannot be processed without a valid pre-existing credit check for the customer.

Segregation of duties to ensure that one individual cannot perform multiple incompatible functions represents an important mechanism for internal control. In computerized environments, segregation of duties is accomplished by means of access controls and authorization matrices that ensure that an authenticated user can only perform functions that he or she is authorised to perform. Gal and McCarthy (1985) demonstrate how the concept of views in a relational database environment can be used to implement segregation of duties. Through properly implemented segregation of duties, controls within the system can ensure that a shipping clerk, for example, cannot create a fictitious sales order to hide the fact that he or she pilfered inventory.

In addition to these controls that each event has in common, each event has its own set of reasonableness/relationship checks that need to be satisfied in order for an instance of the event to occur. In the case of the sales order, the credit limit cannot be null; otherwise, this would mean that the customer has not been duly processed by the credit manager and may be a nonexistent customer, which would cause a fictitious sales order. These reasonableness/relationship checks represent authorized scenarios that create valid sales orders. If these reasonableness/relationship checks fail, then a sales order can still occur if it is approved by an authorized agent, such as the manager (again, segregation of duties).

For brevity, the control ontology has been demonstrated only for the sales order processing subsystem. While, the shipment, collection, and returns and allowances subsystems are not shown, a few points regarding the shipment event are worth making. First, note that in the shipment event, the reliance on separately defined controls for the inventory, customer, and the employees (warehouse person and shipping person) resources and agents would follow the procedure employed in the sales order event. Second, since the sales order event was already validated (as shown in Figure 3), valid data from the sales order can be used to validate instances of shipments, for example to ensure that the shipment date must be greater than sales order date. In sum, the pattern of internal controls shown in Figure 3 can be extended to related and other subsystems.

5 CONCLUSION

This paper described an ontological extension to the REA model to facilitate the identification and documentation of internal controls at the business process level. The proposed approach involves an explicit notation of the audit objectives relevant for each event entity (i.e., business process) on the REA patterned ER diagram (see Figure 2). The proposed extension also involves depicting the detailed controls for the event using UML notation (see Figure 3). The contribution of this paper is in providing a methodology that allows systems analysts, designers, and auditors to document the internal controls that are (and/or should be) present in

the application domain. Such documentation of the internal controls is a key requirement of the Sarbanes-Oxley Act of 2002. We know of no existing conceptual modelling approaches in the literature that provide constructs specifically for documenting the internal controls present (or that should be present) in the application domain.

The internal control ontology presented in this paper needs additional refinement before it can be put to use. We have considered and applied our approach only to the revenue subsystem. The next step will be to extend the ontology to other cycles, such as procurement, conversion, and the inventory cycle to evaluate whether the same pattern holds or whether additional control elements will need to be defined. In future work, it will also be necessary to analyze non-accounting transactions, because while the proposed study has been focused on the financial reporting process, internal controls also have compliance and operational objectives that are also captured in a relational database as part of the enterprise resource system.

Anecdotal evidence suggests that security measures, such as access limitations, are usually not applied at the database level but instead at the application level. Regarding security, applications are often deployed using one of two extreme approaches: (1) with no access rights at all initially and then giving users permissions as needed, or (2) with full access rights initially and then revoking users' permissions as problems are encountered. Both approaches obviously lack the holistic perspective that is necessary to ensure systems integrity. Thus, at the time the system is being designed, it would seem appropriate to model access limitations/rights and to depict such access privileges directly on the entity-relationship diagram to facilitate the incorporation of security controls such as access rights at the table and field level. The modelling approach presented in this paper is a step in the direction of providing analysts and designers with the tools necessary to explicitly indicate user access rights at the time the conceptual data model is being developed.

The internal control ontology proposed in this paper is subject to certain limitations. We acknowledge that it is not sufficient to model internal controls on an individual subsystem basis as demonstrated in Figure 2 and Figure 3. What is needed is a more complete ontology that defines what the controls are within each subsystem and also how the controls in one subsystem relate to other controls in related subsystems. It is also necessary to reconcile our proposed internal control ontology with other ontologies, specifically the Wand and Weber (1989) proposal. Furthermore, rules over how internal controls function in a relational database environment should be developed because, according to Geerts and McCarthy (2000, p. 12), "definition of these rules or axioms is an important part of ontological engineering." The whole area of "ontology" is a very complex area of research. Any axiom definitions will come from analysis and modelling of other accounting and non-accounting cycles. There is a significant degree of overlap between control activities for one business process cycle and other cycles, and also between business process cycles across organizations. Therefore, a generic internal control ontology should be of benefit to analysts and designers responsible for creating accounting information systems. Furthermore, as shown in Figure 3, referential and entity integrity is assumed to occur quite frequently, but this may not be true in some systems. Developing entity-relationship diagrams with internal controls imbued therein should aid in the identification of missing and misspecified internal controls.

One purpose of this study was to instantiate an REA-like model of internal control using UML, as called for by Geerts and McCarthy (2001). However, it is interesting to note that Verdaasdonk (2003), who presented an *ex ante* object-oriented model of an accounting system, suggests that object oriented modelling is not compatible with the REA framework. Further work is necessary to validate or disprove Verdaasdonk's claim.

In conclusion, this study has illustrated an internal control ontology entailing the depiction of internal controls related to audit objectives at the entity-relationship diagram level. Given our focus on accounting information systems, the paper focused on McCarthy's REA framework as the basis for depicting the revenue cycle business processes. Risks, audit objectives, and control procedures for the sales order processing subsystem were illustrated. A UML diagram

of the sales order processing entities was shown, with specific table and field level controls indicated. Future research could use the proposed internal control ontology to investigate whether a developer could design an information system that is deemed to be more "in control" relative to a developer not using the proposed internal control ontology. Future research could also investigate whether an auditor using the internal control ontology could better identify the internal controls that should be present within an accounting information system.

REFERENCES

Arens, Alvin A., and James K. Loebbecke. 1997. *Auditing: An Integrated Approach.* 7th ed. Upper Saddle River: Prentice Hall.

Fox, Mark S., Mihai Barbuceanu, and Michael Gruninger. 1996. An organisation ontology for enterprise modelling: Preliminary concepts for linking structure and behavior. *Computers in Industry* 29:123-134.

Gal, Graham, and William E. McCarthy. 1985. Specification of internal controls in a database environment. *Computers and Security* March:23-32.

Geerts, G. and McCarthy, W. E . 2001. Using Object Templates from the REA Accounting Model to Engineer Business Processes and Tasks. *Working paper* http://www.msu.edu/user/mccarth4/G&M-maintext.htm:1-19.

_____. 2002. An Ontological Analysis of the Primitives of the Extended-REA Enterprise Information Architecture, *The International Journal of Accounting Information Systems* 3(1), pp. 1-16.

Bailey, Andrew D. Jr., Gordon Lon Duke, James Gerlack, Chen-En Ko, Rayman D. Meservy, and Andrew B. Whinston. 1985. TICOM and the Analysis of Internal Controls. *The Accounting Review* 60 (2): 186-201.

COSO Report (Committee of Sponsoring Organizations of the Treadway Commission). 1992. Internal Control - Integrated Framework. New York: AICPA.

ISO/IEC, 2005a. International Standardisation Organisation, Standard 27001: Information Technology - Security techniques- Information Security Management Systems – Requirements (2005-10-15)

_____, 2005b. International Standardisation Organisation, Standard 17799: 2005: Information Technology - Security techniques. Code of practice for information security management, Second Edition (2005-06-15)

Kaplan, David, Ramayya Krishnan, Rema Padmna, and James Peters. 1998. Assessing data quality in accounting information systems. *Communications of the ACM* 41 (2): 72-78.

McCarthy, William E. 1979. An Entity-Relationship View of Accounting Models. *The Accounting Review* LIV (4): 667-686.

———. 1982. The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. *The Accounting Review* LVII (3):554-578.

O'Leary, Daniel E. 1999. Modelling Time In REA/REAL Databases. *Working paper*:1-16.

Sarbanes-Oxley Act. 2002. Public Law No: 107-204. Washington, D.C.: Government Printing Office.

Verdaasdonk, Peter. 2003. An Object-Oriented Model for Ex Ante Accounting Information. *Journal of Information Systems* 17 (1):43 - 61.

Wand, Yair, and Ron Weber. 1989. A Model of Control and Audit Procedure Change in Evolving Data Processing Systems. *The Accounting Review* 64 (1): 87-107.

Weber, Ron A. 2002. Ontological Issues in Accounting Information Systems. In *Researching Accounting as an Information Systems Discipline*, edited by V. Arnold and S. G. Sutton. Sarasota: American Accounting Association.