ACCOUNTING SYSTEM CONCEPTUAL MODELS: THE EFFECT OF ALTERNATIVE MODELS ON AUDITORS' INTERNAL CONTROL REVIEW EFFECTIVENESS AND EFFICIENCY

Mohamed Gomaa, PhD
California State Polytechnic University, Pomona
3801 West Temple Avenue
Pomona, CA 91768
909-869-2374
mgomaa@cpp.edu

Nian Lim (Vic) Lee, PhD
California State Polytechnic University, Pomona
3801 West Temple Avenue
Pomona, CA 91768
909-869-4519
nianlimlee@cpp.edu

Rose Martin-Sheridan, PhD, CPA
California State Polytechnic University, Pomona
3801 West Temple Avenue
Pomona, CA 91768
909-869-02376
rmmartin@cpp.edu

Western Decision Sciences Institute 52nd Conference, April 2-5, 2024

Accounting System Conceptual Models: The Effect of Alternative Models on Auditors' Internal Control Review Effectiveness and Efficiency

ABSTRACT

Auditors' effectiveness and efficiency in examining a company's accounting system will depend on their understanding of the information provided in the system documentation. In addition to information gathered from the client's personnel, this documentation can include flowcharts, system input/outputs, system controls, as well as the conceptual model documenting the system. Understanding the structure of a system and the semantic constraints associated with it is essential for the auditor to accurately assess the control risk associated with that system and to design appropriate audit tests of the client's data. The objective of this paper is to review the literature related to conceptual modeling and its effects on the effectiveness and efficiency of auditors' understanding of the client's information system and controls therein. This paper can serve as a basis for the development of a theoretical foundation for research on the effect of different conceptual model representations on auditors' performance in the review of a client's internal controls.

INTRODUCTION

External auditors need to examine the adequacy and reliability of internal control systems in order to determine the audit evidence to be accumulated. Auditors' effectiveness and efficiency in examining the system will depend on their understanding of the information provided in the system documentation, including document flowcharts, system input/outputs, system controls, as well as the conceptual model documenting the system, in addition to information gathered from the client's personnel. Understanding the structure of a system and the semantic constraints associated with it is essential for the auditor to accurately assess the control risk associated with that system and to design appropriate audit tests of the client's data. Conceptual models, also known as data models or semantic models, are used as an information tool that communicates the structure of the system. These models document the system and the semantic constraints associated with it through diagrammatic representations, such as entity-relationship (ER) diagrams and data flow diagrams (Chen, 1976; and Dunn and Gerard, 2001; Guizzardi et al., 2002)[1]. The technique used to represent the system depends upon the intended user of the conceptual model. For example, models used by designers need to express as much as possible in as much detail as possible, while those used by systems analysts or auditors would contain less detail to be easily readable and understandable (Hay, 1999). The objective of this paper is to review the literature related to conceptual modeling and its effects on auditors' effectiveness and efficiency of examining and understanding a client's information system and controls therein.

Several conceptual modeling approaches to designing information systems have been proposed in the information systems literature. These models are aimed at capturing the logical associations and relationships between data items stored within the organization's information system, which represent its information needs (Murthy and Wiggins 1993). The early conceptual models include hierarchical models (e.g. Colantoni, Maines and Whinston (1971), Lieberman and Whinston (1975), Haseman and Whinston (1976)), network models (e.g. Haseman and Whinston (1976)), and relational models (e.g. Everest and Weber (1977)). These modeling approaches were limited in their ability to only represent those semantics that could be implemented on the target database management system. The data representation capabilities of these models can be very limiting to both the designers and users of the system, given the complexities of the real world.

The Resources-Events-Agents (REA) Accounting Model developed by McCarthy (1982) proposes the use of an accounting framework whose structure is derived with semantic modeling and whose elements correspond closely to the ideas of Yuji Ijiri (1975), Mattessich (1964), Buckley and Lightner (1973), and Yu (1976). Using this modeling approach, a large portion of the semantics of the application are captured and the conceptual modeling of enterprise-wide schema is facilitated. As noted by Nakamura and

[1] Rolland and Proix (1992) propose a linguistic approach to conceptual modeling using natural language statements, based on Filmore's (1968) "CASE for CASE" theory, during the development stage of a database system to assist systems developers by providing a conceptual specification of the system.

Johnson (1998, p. 3), “most existing systems based on the REA model heavily depend on relational database technology”.

Several studies have compared relational models with other representations (e.g. Batra et al 1990; and Batra and Anthony 1994). Relational database technology is subject to some limitations that constrain designers’ ability to represent the semantics of complex systems and affect the effectiveness and efficiency of these systems. One of the limitations of using relational database technology is its inability to support information hiding except through password protection (Nakamura and Johnson 1998). Therefore, everything your program needs becomes public to your program. In object-oriented programming, encapsulation refers to the bundling of data and the methods that work on that data together as a single unit. Encapsulation provides a superior means of incorporating internal controls over accounting procedures (Adamson 1993, p. 19; Chu 1992). As noted by Adamson (1993), encapsulation permits each class’s interface to reveal as little as possible about the class’s inner workings. In other words, users only know that values are being passed and processed, but do not know how classes are being implemented or stored. In contrast, using relational database technology, the “attributes of each entity are accessible by other entity types,” reducing internal control over accounting procedures. A second limitation is that relational databases cannot store functions to realize inheritance[2]. Relational databases can only realize attribute (data) inheritance and will not enable the inheritance of behaviors (functions) (Adamson 1993). For example, in an accounting context, an entity type, ‘Notes Receivable’, that is interest bearing, may have an IS-A[3] relationship with an entity ‘Receivables’ (Adamson, 1993). The ‘Receivables’ entity may contain functions for calculating interest. However, in a relational database,

[2] Inheritance is a core concept of object-oriented programming. When a new object is created, inheritance allows the object’s properties to be the same as the existing class that it belongs to (i.e., an object inherits all of the properties of its class), and only those properties that differ from class properties will need to be declared (Taivalsaari, 1996).

[3] An IS-A relationship is complete inheritance. In this type of relationship, the child class (class that inherits) is defined as a type of the parent class (class that is inherited from) and inherits all properties of the parent class.

the ‘Notes Receivables’ entity can only inherit the attributes (but not functions) from the ‘Receivables’ entity. Thus, a relational database would have to redefine the interest calculation function resulting in unnecessary redundancy.

To overcome these limitations, many researchers have suggested the object-oriented approach as a better approach to designing an information system. Object-oriented databases provide a high level of congruence between the data model for the application and the data model for the database (McClure 1997). This high level of congruence leads to more natural data structures, better manageability, and higher system reliability and security. One of the advantages of using object-oriented database technology is its ability to encapsulate data and procedures within an object.[4] This would allow the interface of each class to implement details of an object, providing a superior means of incorporating internal control over accounting procedures (Adamson 1993; Chu 1992).[5] In addition, object-oriented databases support inheritance, enabling subclasses that are derived from superclasses to be stored with a single operation, reducing redundancy and enhancing efficiency. Inheritance also permits the creation of new objects that automatically inherit the attributes and methods of an existing object by referring to it, which enhances the expandability of the system (Murthy and Wiggins 1993) and reduces the chances of errors in the assignment of attributes. Using the ‘Receivables’ example used above, a superclass ‘Interest Bearing’ could be defined to contain the interest calculation method. The ‘Receivables’ object and the ‘Notes Receivables’ object would inherit the interest calculation method from the ‘Interest Bearing’ object, eliminating the need to redefine this method, therefore reducing redundancy and increasing efficiency. In addition, object-

[4] Data encapsulation is a fundamental part of Object-Oriented programming. It refers to the bundling of the data with the methods used to work with the data. It effectively hides the state of the data inside a class, preventing unauthorized access to it.

[5] A class is defined as a specific category of related data (e.g. airplanes, plants, etc.). For example, a ‘Tree’ class might include two subclasses, ‘CommonName’ and ‘ScientificName’, as well as a ‘GetTreeInfo’ method. A class can also have subclasses (members) with limited access. For example, it can be classified as Private or Protected, rather than Public. These classifications can also be applied to an entire class.

oriented database systems support polymorphism[6], which allows users to define their own data types, and facilitates the specification of unique operators and functions that manipulate user-defined data types (Murthy and Wiggins 1993), something that is not possible with relational databases.

Several researchers have studied the effectiveness and efficiency of using different representations in problem solving (e.g. Vassey 1991; Smelcer and Carmel 1997; Morris et al. 1999; Dunn and Gerard 2001; Dunn and Grabski 2001). Most of this research utilized the theory of cognitive fit developed by Vassey (1991) to explain performance differences among users of alternative models. The notion behind this theory is that problem solving aids, such as problem representations, will effectively reduce the complexities in the task environment when they support the methods or processes required to perform the task (Vassey 1991). Therefore, if there is a correspondence between the task and the method of information presentation, the effectiveness and efficiency of the task performance of individuals will be increased. The objective of this paper is to review the literature related to conceptual modeling and its effects on the effectiveness and efficiency of auditors' understanding of the client's information system and controls therein. This paper can serve as a basis for the development of a theoretical foundation for research on the effect of different conceptual model representations on auditors' performance in the review of a client's internal controls.

The remainder of this paper is organized as follows. Section 2 provides a review of the literature related to systems modeling. Section 3 provides a review of prior studies investigating techniques used by auditors to obtain an understanding of the client's accounting systems and controls therein and the effects of alternative representations on the effectiveness and efficiency of this understanding. Section 4 discusses implications for auditors' performance in evaluating internal control systems. Finally, section 5 provides concluding remarks.

[6] Polymorphism allows the same program code to work with multiple data types. This improves simplicity and code maintenance.

LITERATURE REVIEW OF SYSTEMS MODELING

Semantic data models attempt to communicate the structure of a system and its underlying constraints. As noted by Dunn and McCarthy (1997), Colantoni, Manes and Whinston (1971) were the first accounting researchers to connect database technology with the problem of building more powerful disaggregated and multidimensional database technology. Colantoni et al, (1971), Lieberman and Whinston (1975), and Haseman and Whinston (1976) discussed implementation of Sorter's "events" accounting theory into hierarchical database models.[7] However, these modeling approaches were limited in their ability to only represent those semantics that can be implemented on the database management system. Codd (1970) conceptualized a relational model that represents data in terms of two-dimensional tables, and represents the links between the tables by means of record keys. Everest and Weber (1977) applied some concepts of Codd's (1970) relational database model to overcome the hierarchical model's lack of data independence.

Several modeling approaches were proposed in the late 1970's and early 1980's that attempted to capture more of the semantics of the data[8] (e.g., Chen 1976; Hammer and McLeod 1981; Hull and King 1987). McCarthy (1979) integrated accounting theory with database systems modeling demonstrating how the entity-relationship approach could be applied to modeling information requirements in an accounting context (McCarthy 1979; Murthy and Wiggins 1993). In 1982, McCarthy expanded his earlier work and proposed

[7] In 1969, Sorter proposed a new orientation for accounting theory, which he called an "events accounting" approach, as a solution to the problem with the conventional accounting approach, which he called the "value theory" approach. Using this approach, accounting should "provide information about relevant economic events that allows individual users to generate their own individual decision models [because] the loss of information generated by aggregation and valuation by the accountant is greater than the associated benefit" (Sorter, 1969, p.13). Using events theory, the entity's accountants observe and record relevant economic events, then report these event-observations to the users who are free to individually aggregate and evaluate the data and use the resulting information in decision making (Cushing 1989).

[8] The relationships between the data objects (e.g. the relationship between a Customer in a Customer table and a Sales Transaction in a Sales table).

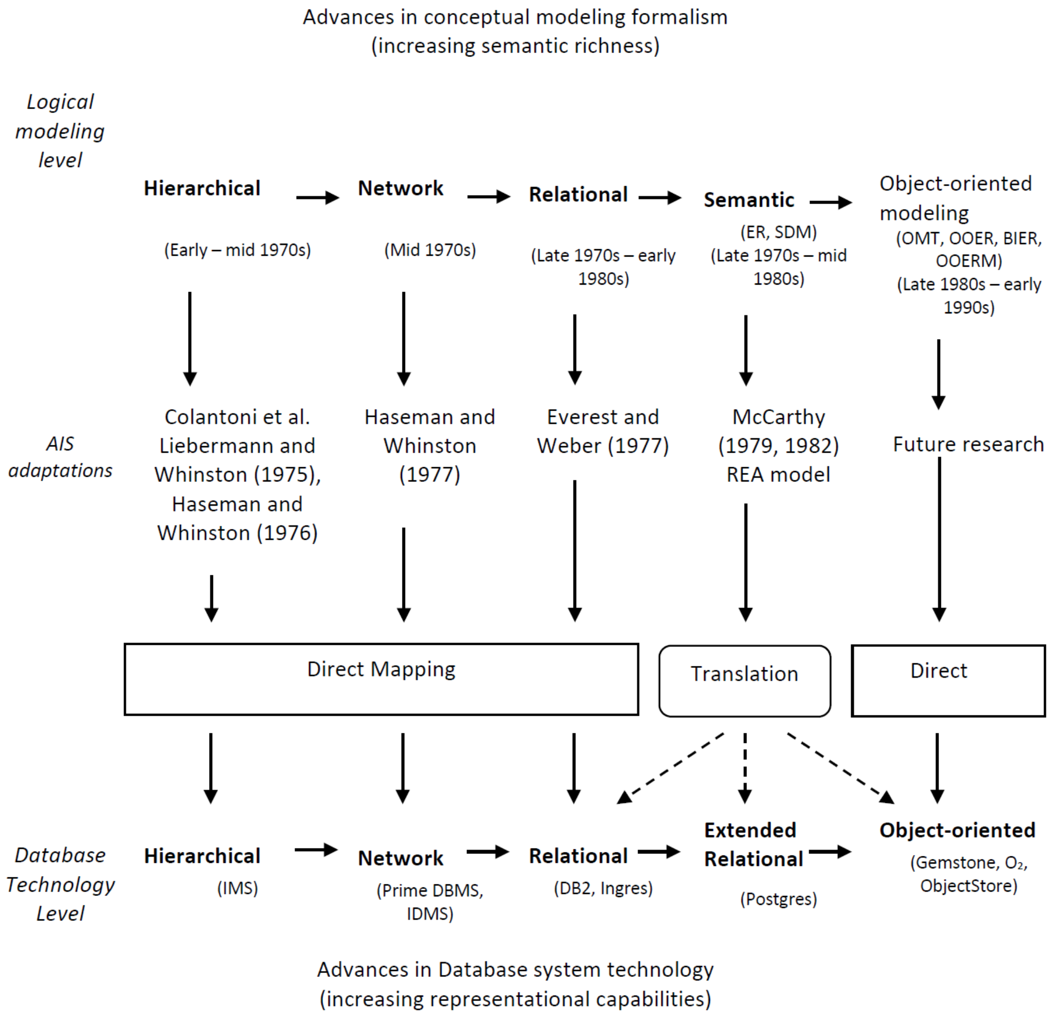
the REA accounting framework (McCarthy 1982). This framework includes the concept of generalization proposed by Smith and Smith (1977)[9] and provides a means of representing information about economic resources, economic events, and economic agents as well as the relationships between them (McCarthy, 1982; Murthy and Wiggins, 1993). This modeling approach captures much of the semantics of the application and facilitates the conceptual modeling of enterprise-wide schema.

The primary limitation of these modeling approaches is that they had to be translated and implemented on available database management systems which were not capable of fully representing all the semantics expressed in the model (Murthy and Wiggins 1993). Figure 1 illustrates the progression of data modeling research, the accounting adaptations, and the database technology, and indicates whether the logical model requires translation or can be directly mapped into a particular database.

[9] Smith and Smith (1977) define generalization as "an abstraction in which a set of similar objects is regarded as a generic object." For example, students at a university can be abstracted to a generic object "Student." This generic classification ignores differences that students may have such as name, age, major, etc.

Figure 1

Progression of Database Modeling Research and DBMS Technology
(adapted from Murthy and Wiggins (1993))


Advances in conceptual modeling formalism
(increasing semantic richness)
Logical modeling level
Hierarchical
(Early – mid 1970s)
Network
(Mid 1970s)
Relational
(Late 1970s – early 1980s)
Semantic
(ER, SDM)
(Late 1970s – mid 1980s)
Object-oriented modeling
(OMT, OOER, BIER, OOERM)
(Late 1980s – early 1990s)
AIS adaptations
Colantoni et al. Liebermann and Whinston (1975), Haseman and Whinston (1976)
Haseman and Whinston (1977)
Everest and Weber (1977)
McCarthy (1979, 1982) REA model
Future research
Direct Mapping
Translation
Direct
Database Technology Level
Hierarchical
(IMS)
Network
(Prime DBMS, IDMS)
Relational
(DB2, Ingres)
Extended Relational
(Postgres)
Object-oriented
(Gemstone, O₂, ObjectStore)
Advances in Database system technology
(increasing representational capabilities)

REA Data Model

McCarthy (1982) extended his entity relationship approach to explore the issues of database design in a larger organizational context. With the REA model, McCarthy expanded the entity – resource (E-R) accounting methodology to include the concept of generalization hierarchies proposed by Smith and Smith (1977). This model provides a generalized E-R representation of accounting phenomena that would facilitate the conceptual modeling of enterprise-wide schema. McCarthy (1982) saw that the development process of a database schema for complex organizations requires four steps. First, the requirements of the information users are analyzed. Second, the information needs of the users are modeled using a semantic data model (called view modeling)[10]. Third, an enterprise-wide conceptual schema is developed by integrating the differing information needs of the users. Fourth, an internal schema is developed from the conceptual schema for final application. Following this logic, McCarthy developed the REA model to include three essential elements: First, a set of rules for determining which elements in the entity-wide database are within the accounting domain; Second, a graphical modeling representation of the business processes in the accounting domain; Third, a set of techniques for the construction of corporate databases following the first and second elements (Debreceny 1998).

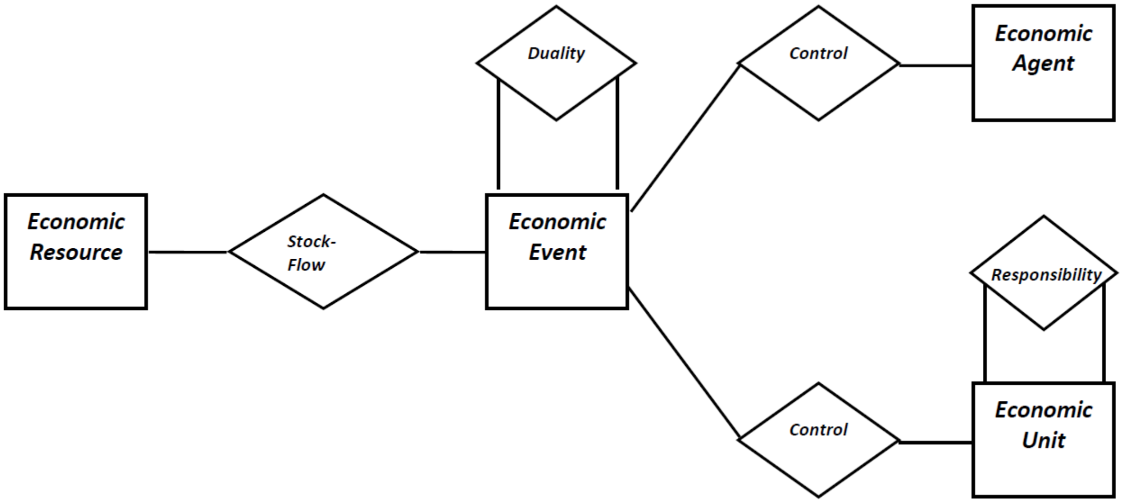
McCarthy (1982) divides the REA model into two structural elements. First, those elements related to economic resources and events. Second, those elements related to economic agents. McCarthy draws his definition of economic resources from Ijiri (1975) as “objects that are scarce, have utility and are under the control of an enterprise” (McCarthy 1982, p. 562). As cited by McCarthy (1982, p. 562), Yu (1976, p.256) defines economic events as “a class of phenomena which reflect changes in scarce means [economic resources] resulting from production, exchange, consumption, and distribution.” The REA model links economic resources with economic events using stock-flow relationship.[11] McCarthy drew the concept of duality from Ijiri (1975) and Mattessich (1964) to be the relationship that links each increment in the resource set of the enterprise with a corresponding decrement. Economic agents include external agents such as the organization’s customers or suppliers, and internal agents such as the

[10] A view model provides a presentation of a system from the viewpoint of various perspectives, such as the viewpoints of different categories of users of a system.

[11] A stock-flow relationship refers to the relationship between an economic resource (stock) and an economic event (flow). For example, this could be the relationship between inventory as an economic resource and sales as an economic event.

organization's managers. Figure 2 illustrates the relationships between the resources, events and agents within the model.

Figure 2
The REA Accounting Model



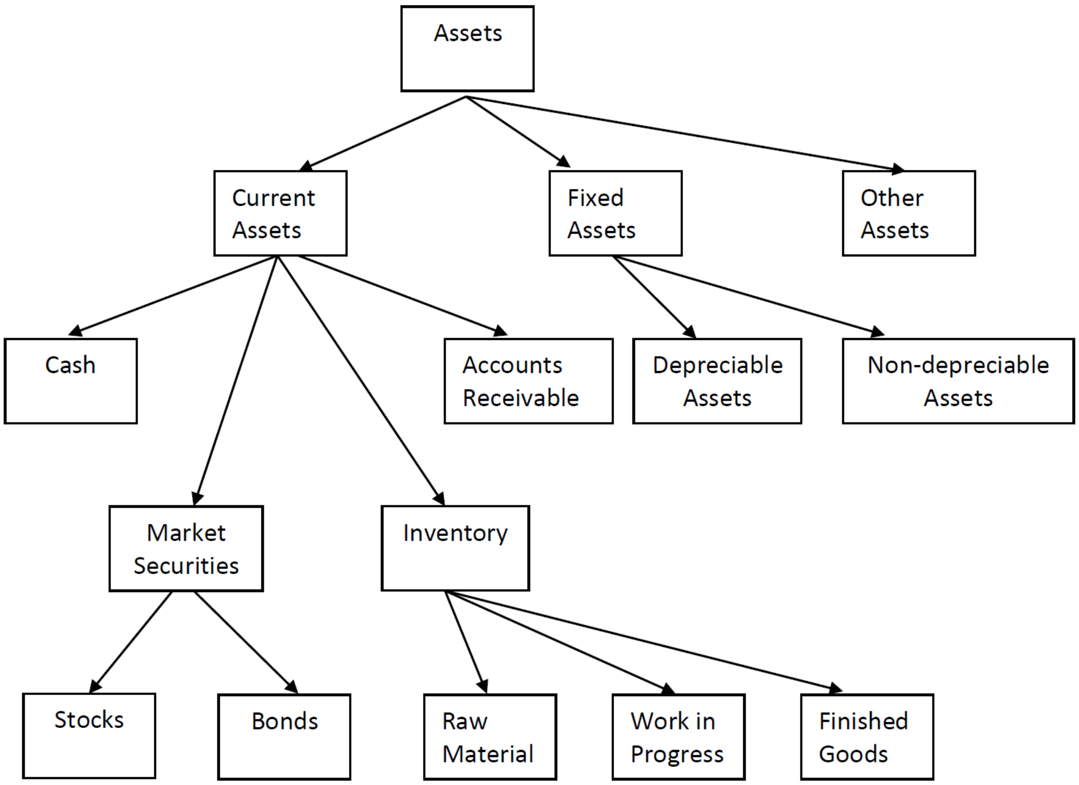
Most of the existing systems based on the REA model depend heavily on relational database technology and tend to be inefficient (Nakamura and Johnson 1998). Relational database technology is not able to support information hiding except through password protection (Nakamura and Johnson 1998). Relational databases cannot store functions to realize inheritance. A relational database can only realize attribute inheritance and will not enable the inheritance of behaviors (Adamson 1993). As discussed earlier in the paper, this would result in storing items several times in different tables causing unnecessary redundancy and complicating the access of these data items. To overcome these limitations, many researchers have suggested the object-oriented approach as a better approach to designing an information system.

Object-Oriented Modeling

As noted by Debreceny (1998), object-orientation incorporates a number of interrelated and overlapping technologies including programming languages (Coad and Nicola 1993; Lorenz 1993), data models (Bancilhon 1996; Kim 1990; Khoshafian 1993; Cattell 1994; Loomis 1995), methodologies (Holland and Lieberherr 1996; Graham 1994, 1995; Martin and Odell 1995; Booch, Rumbaugh, and Jacobson 1998), and user interfaces (Khoshafian and Abnous 1990). This paradigm has several important and powerful properties that differentiate it from other paradigms. One of these properties is abstraction, which provides the ability to group objects with the same characteristics (classes). Another property, encapsulation, provides the ability to differentiate between the external interface of the object from the internal implementation of the object by hiding the internal implementation.[12] Inheritance is another fundamental principle of the object-oriented paradigm. Using inheritance an object would inherit the attributes and methods from objects that are higher in the same branch of the hierarchy. Another principle of the object-oriented paradigm is polymorphism. It allows different objects to respond to the same message in different ways.

[12] For example, an "Order" as an object in an object-oriented system can contain all of the information about an order (e.g. customer, item, delivery date, etc.) as well as the "method" required to compute the amount of the order.

Figure 3
The hierarchy of asset accounts
(adapted from Chu (1992))



Using object-oriented modeling, classes are arranged to support attribute inheritance. In applying this to an accounting system, accounts are organized by accountants into hierarchies (Chu, 1992). Using object-oriented concepts, the account categories in these hierarchies can be modeled as classes. Figure 3 illustrates the hierarchy of asset accounts. This figure shows the class relationships between the asset accounts categories. In this case, each class is a superclass to the classes below it. For example, “Current Assets” is a subclass of the superclass “Assets” and is a superclass to the subclass “Cash”. This classification, suggested by Chu (1992), uses the chart of accounts as a basis for constructing object classes. This limits the model development as the chart of accounts does not capture all aspects that directly impact an organization (Murthy and Wiggins 1997).

The events associated with these object classes also need to be identified. Using the events accounting approach, recording of the data corresponds to the occurrence of economic events affecting a firm (Lieberman and Whinston 1975). For example, using the events approach, a sales event would result in the generation of revenue and the sale of an asset. This sales event would be an object that inherits the attributes of the superclass "Sales Transaction" and would affect the objects it is associated with, such as the inventory object. Kandelin and Lin (1992) developed a computational model of an events-based object-oriented accounting information system consisting of three main components: an event message database system, an accounting object report system, and an accounting intelligence subsystem (Murthy and Wiggins 1997). They did not, however, develop a logical model of an events-based object-oriented model nor did they present a methodology for designing one (Murthy and Wiggins 1997). Murthy and Wiggins (1997) present a modeling approach for designing business information systems that focuses on business events as a basis for identifying objects and the inter-relationships between them. Their model captures both the structural aspects and the behavioral aspects of modeling a business.

Other researchers have studied various aspects of the object-oriented methodology. Wang (1996) studied the effectiveness and efficiency of the object-oriented systems development approach to develop a practical systems analysis method in the object-oriented paradigm. Chuang and Yadav (2000) proposed a decision-driven approach to object-oriented analysis. Their approach consists of two phases: a top-down understanding of the organization and a bottom-up specification of information requirements. Nagarur and Kaewplang (1999) developed an object-oriented decision support system to assist in maintenance planning. They used object-oriented databases and relational databases for transient and permanent entities respectively. Chen et al. (1995) present a knowledge-based approach and mechanism to support dynamic accounting database schema evolution in an object-based data-modeling context. Smith and Zdonik (1987) studied the differences between object-oriented databases and relational databases examining the ramifications of replacing the relational database with an object-oriented database on a hypermedia[13] system implemented on a relational database management system. They found that the object-oriented database could successfully handle the data requirements of the hypermedia system. Shoval and Shiran (1997) compared extended entity relationship (EER) models and object-oriented models from the point of view of design quality. They used three measures for quality: correctness of designed conceptual schema, time needed to complete the design task, and the designers' model preferences. They found that designers generally prefer the EER model and it generally takes them less time to design an EER schema.

As noted by Adamson (1993), several researchers have shown that there are a number of advantages to using the object-oriented model. First, the object-oriented model allows for simplification of system design and the modification of program segments using objects because all of the procedures that depend on the object's representation are localized (O'Keffe 1986). It also allows designers to disregard objects when designing or revising segments, and create methods without completely specifying the structure and behavior of every class of entities (Atkinson et al. 1990). Object-oriented models thus ensure the integrity of an object's private information since applications cannot modify object data directly (Maier and Stein 1987), increasing the efficiency of procedures that operate on objects since these procedures can be designed to deal with specific data types rather having to accommodate many different typess (O'Keffe 1986). Object-oriented models also increase the number of characteristics that can be inherited, making the object model more powerful than the semantic data model (Navathe 1992). As a result, thinking in terms of objects is one way of organizing or abstracting the real world, which would result in a model that closely matches reality (Booch 1986).

[13] Hypermedia is a medium of information that includes graphics, audio, video, plain text, and hyperlinks that may also be interactive in nature (Nelson, 1965)

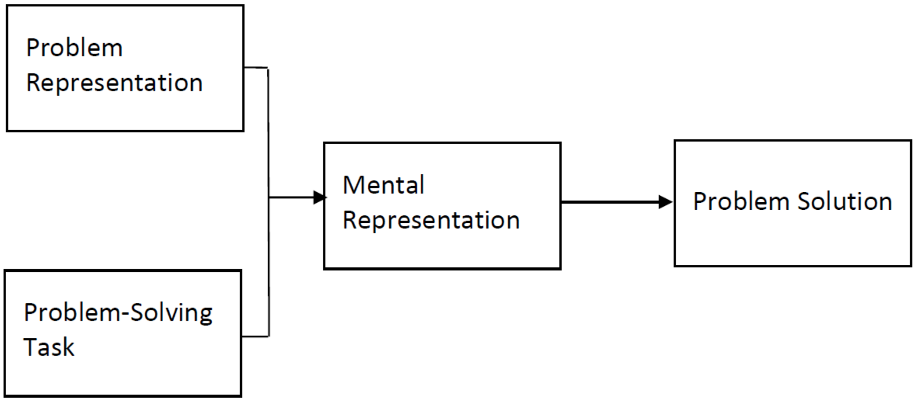
Auditors' Understanding of Accounting Systems and their Controls

Auditing standards require external auditors to examine the adequacy and reliability of internal control systems in order to determine the audit evidence to be accumulated. The essence of internal control is to develop a system of checks and balances that ensures that all the events and activities that occurred in an organization are in accordance with organizational objectives and adhere to management policies and procedures. In an auditing engagement where most of a client's data are in electronic format, auditors used to audit around the computer by scrutinizing printouts of the important information provided by the system (Lanza 1998). However, as businesses are increasingly moving towards more complex systems and relying more on electronic data, auditors need to audit through rather than around the computer. They will need to access the integrity of the system and gain a better understanding of the system as a whole. Database integrity is the primary tool of database auditing (Davis and Weber 1986; Orman 1997). Several software packages have been developed to assist auditors in examining the client's data. ACL Software and Interactive Data Extraction & Analysis (IDEA) are the two leading auditing software packages (Lanza 1998). These packages not only improve the audit effectiveness by enabling the auditor to test 100% of the client's data, they also improve auditors' understanding of the client's business by giving them a broader view of the enterprise through the effective inspection of large volumes of the client's data (Lanza 1998). However, these software packages don't provide auditors with an understanding of the structure of the client's underlying system.

The integrity of the client's database is a major factor in determining the control risk of an audit engagement. Orman (1997) studied three major strategies of database auditing and compared them in terms of efficiency and effectiveness in determining errors. The first is periodic enforcement in which all integrity constraints are executed at the end of a fixed period, and in which errors and violations that have occurred during that period are caught and corrected. Integrity constraints are logical statements that capture the semantics of data and the intent of a transaction. The second approach is a transaction counting approach in which the constraints are executed after every n transactions. A special case of this approach is real-time continuous integrity enforcement. The third approach is a variation of real-time continuous enforcement in which integrity constraints are executed after each transaction, but not in their full and complete form. A simplified version of each integrity constraint is executed to test if a new error was introduced into the database by a given transaction (Motro 1989). Orman (1997) found that hybrid strategies outperformed pure strategies under certain conditions.

In order to effectively inspect the system, auditors need to develop a better understanding of the system. The method of system documentation used by the client affects how an auditor understands the system and performs the required auditing task. The different conceptual/semantic models used by the client to document the system can improve or reduce the effectiveness and efficiency of the auditor. Several studies have looked at the effect of different representation schemes on decision-making and problem-solving performance (e.g. Smelcer and Carmel 1997; Morris et al. 1999; Dunn and Gerard 2001; Dunn and Grabski 2001). Most of these studies used the cognitive fit model developed by Vassey (1991) to provide an explanation for performance differences between individuals presented with different representations.

Figure 4
The general problem-solving model of cognitive fit
(adapted from Vassey (1991))



Vassey's (1991) theory of cognitive fit posits that a correspondence between the data representation format and the task leads to superior task performance by individuals. Mental representation is the way in which an individual represents a problem in working memory using the characteristics of both the problem representation and the problem-solving task. Using this notion, Vassey predicted that when the type of information presented in the problem-representation and the problem-solving tasks match, an individual would formulate a mental representation that emphasizes the same type of information, leading to effective and efficient problem-solving performance. However, when there is a mismatch between the problem representation and the problem-solving task, individuals will need to transform the data derived from the problem presentation in order to perform the task leading to reduced performance.

Applying the theory of cognitive fit to accounting models, Dunn and Grabski (2001) examine how cognitive fit works based on the concept of localization, proposed by Larkin and Simon (1987), using alternative accounting models. Larkin and Simon noticed that if a representation allowed individuals to direct their attention to a limited area, meaning “localization” of focus, their performance will be better (Dunn and Grabski 2001). Dunn and Grabski (2001) use two accounting model representations in their comparative study, the traditional debit-credit-account accounting model (DCA) and the resources-events-agent accounting model (REA). Using experienced and novice participants provided with the same information in differing formats, they found that localization is an important element of cognitive fit. This suggests that representations with a high degree of localization, such as the REA model, can direct an individual’s attention enough to improve performance.

In studying the effect of different representations on auditor effectiveness and efficiency, Dunn and Gerard (2001) examined the difference between the entity-relationship model (ER) and the Backus-Naur Form (BNF) grammars. Both the ER model, which is a diagrammatic representation, and the BNF model, which is a linguistic representation, are informationally equivalent (Dunn and Gerard 2001). They used auditors to perform tasks involving information searches, recognition and inference using the revenue cycle and the acquisition cycle. The ER diagrams were based on McCarthy’s (1982) REA model. They found no difference in terms of accuracy for auditors using the two representations. However, the ER diagrams were easier to use and they provided significant computational efficiency over the BNF grammar representation (Dunn and Gerard (2001).

In another study, Morris et al. (1999) examined both experienced and novice systems analysts using both procedural and object-oriented techniques. They used a human problem-solving model developed by Newell and Simon (1972), which is similar to the theory of cognitive fit, to examine the differences between the two techniques. They found that both novice and experienced subjects demonstrated higher subjective mental workload when using object-oriented analysis. They also found that novices prefer object-oriented techniques and find these techniques easier as compared to experienced subjects. These studies and others examining the differences between various modeling techniques have mixed results as to the superiority of one modeling technique over the other pointing out the need for further research in this area.

Implications for Auditor's Performance in Evaluating Internal Control Systems

The above review has important implications on the study of auditor's performance when studying and evaluating internal control systems. In order for the auditors to be able to evaluate an internal control system, they need to develop a better understanding of the structure of the system and its constraints. As indicated in Section 2 of this paper, there are several data modeling approaches to designing information systems including hierarchical models, network models, relational models, and object-oriented models. The data representation capabilities of systems developed using these modeling approaches vary considerably. Modeling approaches use different methods of system documentation that might have different effects on the auditors' understanding of the system and their performance when evaluating the system.

Studies reviewed in Section 3 examine the effect of different representation schemes on the performance of decision makers and problem-solving performance. Most of these studies employ the cognitive fit model developed by Vassey (1991) to provide an explanation for performance differences when individuals are exposed to different representations. The results of these studies lead to the following conclusions: First, performance of individuals will be better when a representation allows them to direct their attention to a limited area (Dunn and Grabiski 2001). Second, a diagrammatic representation is easier to use and provides significant computational efficiency over a linguistic representation, but there is no difference in accuracy for auditors using the two representations (Dunn and Gerard, 2001). Third, Individuals demonstrate higher mental workload when using object-oriented analysis compared to procedural analysis (Morris et al, 1999).

Dunn and Gerard (2001) were among the first to examine “conceptual models of enterprise databases as used by professional auditors” (p. 224). They recognize however, that theory in the area of the effect of different representations on individual's performance "appears to be underdeveloped"(p.245), and more research needs to be done in that area. Their study lays the foundation for much future research in the accounting and auditing domain. A natural extension to the Dunn and Gerard (2001) work is to examine the effect of alternative conceptual model representations of the accounting system on auditors’ internal control review. In order for auditors to effectively review a client's system, especially in a highly computerized environment, they need to better understand the structure of the system and its constraints. The method of system documentation used by the client will directly affect this and in turn their audit performance. A better understanding of the system will likely be reflected in less time in internal control review (efficiency) and more accurate assessment of control risk (effectiveness).

To investigate whether different system representations have different effects on the performance of the auditor' review of the internal control system, the general framework developed by Larkin and Simon (1987) and used by Dunn and Gerard can be utilized. In Larkin and Simon's framework, "the cognitive computational efficiency of a representation depends on three factors: data structure (representation), programs, and attention management system, and how well those three factors work together."(Dunn and Gerard, 2001, p. 228). The programs use three processes to solve problems; search, recognition, and inference. As described in Dunn and Gerard (2001), "search is the process that requires attention management…localization as an attention manager may reduce search space considerably" (p. 228). "Recognition is even more sensitive to representation format, as it involves first searching and then matching information in the representation of information in the problem statement" (p. 228). Inference, on the other hand, is less "susceptible to representation effects as long as the representations are informationally equivalent" (p. 228). Based on this framework, Dunn and Gerard (2001) conclude that diagrammatic form of representation will result in better performance.

In the internal control review context, the search and recognition processes are expected to be the most affected by system representation. Since different conceptual models use different representations, it is expected that when an auditor reviews different systems, his performance, in terms of the time spent to perform the task and the accuracy of control risk assessment, will be different. Dunn and Gerard (2001) examined the effects of an ER diagrammatic form compared to a linguistic form. However, they recognize that there are numerous "types of system documentation" (p.224). As an extension to their study, one might try to investigate the effect of different types of diagrammatic representation (e.g., ER vs. object-oriented representations) on auditors' performance in the internal review process. System documentation when object-oriented approach is used is different of that in other traditional systems. Existing literature suggests that the object-oriented model is superior in some aspects and inferior in some aspects to the traditional models as indicated in Sections 2 and 3 of this paper. Therefore, we cannot develop a directional proposition with respect to the effect of representations on the efficiency of internal control review. If the efficiency of auditors' performance is measured in terms of the time spent in performing internal control review, and the effectiveness is measured by the accuracy of control risk assessment, one can formulate the following propositions:

Efficiency of internal control review:

Proposition 1: There is a difference in the time spent by the auditor in the internal control review when object-oriented representation is used compared to the ER representation.

Using the Larkin and Simon's framework and following Dunn and Gerard, this general proposition can be stated as follows:

Proposition 1a: There is a difference in the time spent on the search task by the auditor in the internal control review when object-oriented representation is used compared to the ER representation.

Proposition 1b: There is a difference in the time spent on the recognition task by the auditor in the internal control review when object-oriented representation is used compared to the ER representation.

Proposition 1c: There is a difference in the time spent on the inference task by the auditor in the internal control review when object-oriented representation is used compared to the ER representation.

Effectiveness of internal control review:

As noted by Rob and Coronel (1999), the object-oriented approach provides a natural structure environment that strengthens the internal control system through the encapsulation of the properties and behavior within the object and prohibiting unauthorized personnel from accessing the system. In contrast, the traditional system's data components or fields are directly accessible from the external environment (Rob and Coronel, 1999).[14] Thus, it is expected that internal control systems developed under traditional approaches (e.g., E-R, Relational) are more vulnerable to unauthorized access than those developed under object-oriented principles. Therefore, the control risk (and hence audit risk) involved in object-oriented information systems are likely to be lower than the control risk involved in traditionally developed information systems. Since both approaches use diagrammatic representation, it is expected that there will be no differences in attention direction mechanisms, and therefore, it is not expected to result in significant differences in accuracy. This leads to the following Proposition:

Proposition 2: Auditors' assessment of control risk with object-oriented representations will not be significantly different from that made with ER representation.

SUMMARY AND CONCLUSION

Understanding the structure of client's system and the semantic constraints associated with it is essential for the auditor to more accurately assess the control risk associated

[14] Rob and Coronel (1999, p.503) point out that the closest thing to a method in traditional approaches is the use of triggers and stored procedures in SQL database. They argue further that because triggers do not include the encapsulation and inheritance benefits, they do not yield the same functionality as methods.

with the system. Different conceptual models have been discussed in the information systems literature. These models are used to document the system and used as an information tool that communicates the structure of a system and the semantic constraints associated with it. It is argued in the literature that different conceptual models provide different representations. Understanding the information content of these representations is important for the auditor to better understand the system and more accurately assess the control risk.

Several studies have addressed the issue of whether different conceptual model representations affect the performance of the users of these models. The results of these studies, however, are mixed. For example, while Dunn and Gerard (2001) find that conceptual model representation has effect on individual's performance, McCarthy (1982) finds no difference in terms of accuracy for auditors using two different representations. They, however, find that one representation (ER diagrams) provides significant computational efficiency over BNF grammar representation. This paper presents a review of the literature related to systems modeling and their effects on the effectiveness and efficiency of individuals' understanding of systems. This review establishes a theoretical foundation for research on the effect of different conceptual model representations on auditor's performance.

REFERENCES

Adamson, Ian L. (1993), "An object-oriented financial accounting information system", Published PhD. Dissertation, University of Waterloo.

Atkinson, M., F. Bancilhon, D. DeWitt, K. Dittrich, D. Maier, and Zdonik, S. (1990), "The object-oriented database system manifesto", In *Deductive and Object-Oriented Databases*, edited by W. Kim, J. Nicolas and S. Nishio. Amsterdam: Elsevier Science Publishers.

Bancilhon, F. (1996), "Object databases", *ACM Computing Surveys (CSUR)*, Vol. 28 No. 1, pp. 137-140.

Batra, D., J. Hoffer, and Bostrom, R. (1990), "Comparing representations with relational and EER models", *Communications of the ACM*, Vol. 33 No. 2, pp. 126-139.

Batra, D., and Anthony, S. (1994), "Effects of data model and task characteristics on designer performance: a laboratory study." *International Journal of Human-Computer Studies* Vol. 41, pp. :481-508.

Booch, G., (1986), "Object-Oriented Development", *IEEE Transactions on Software Engineering*. Vol. 12 No. 2, pp. 211-221.

Booch, G., Rumbaugh, J. and Jacobson, I. (1998), "The unified modeling language users guide", Addison-Wesley Professional.

Cattell, R. (1994), "Object data management: object-oriented and extended 1st", Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA

Chuang, T, and Yadav, S. B. (2000), "A decision-driven approach to object-oriented analysis." *Database for Advances in Information* Systems, Vol. 31 No. 2, pp. 13–34.

Chen, P. (1976), "The entity-relationship model: Toward a unified view of data." *ACM Transactions on Database Systems,* Vol. 1 No. 1, pp. 9-36.

Chen, J., D. McLead, and O'Leary, D. (1995), "Domain-Knowledge-Guided Schema Evolution for Accounting Database Systems", *Expert Systems with Applications*. Vol. 9 No. 4, pp. 491-501.

Chu, P. C. (1992), "An object-oriented approach to modeling financial accounting systems", *Accounting, Management and Information Technology* Vol. 2 No. 1, pp. 39-56.

Coad, P., and Nicola, J. (1993), "Object-oriented programming", Englewood Cliffs, NJ: Yourdon Press.

Colantoni, C. S., R. P. Manes, and Whinston, A. (1971), "A unified approach to the theory of accounting and information systems", *Accounting Review* Vol. 46 No. 1, pp. 90-102.

Cushing, B. E. (1989), "A Kuhnian interpretation of the historical evolution of accounting." *The Accounting Historians Journal,* Vol. December, pp. 1-41.

Davis, G. B., and Weber, R. (1986), "The Impact of Advanced Computer-Systems On Controls And Audit Procedures-A Theory and an Empirical-Test." *Auditing: A Journal of Practice & Theory* Vol. 5 No. 2, pp. 35-49.

Debreceny, R. (1998), "Object and Relational Database Accounting Information Systems:
An Experimental Evaluation", PhD. Dissertation, Southern Cross University.

Dunn, C., and McCarthy, W. E. (1997), "The REA accounting model: Intellectual heritage and prospects for progress", *Journal of Information Systems* Voll. 11 No. 1, pp. 31-52.

Dunn, C., and Gerard, G. (2001), "Auditor efficiency and effectiveness with diagramatic and linguistic conceptual model representations", *International Journal of Accounting Information Systems* Vol. 2, pp. 223-248.

Dunn, C., and Grabski, S. (2001), "An Investigation of Localization as an Element of Cognitive Fit in Accounting Model Representation." *Decision Sciences* Vol. 33 No. 1, pp. 55-94.

Everest, G. C., and Weber, R. (1977), "A Relational Approach to Accounting Models." *Accounting Review* Vol. 52 No. 2, pp. 342-359.

Graham, I., (1994), "Object-Oriented Methods", London: Addison-Wesley

Graham, I. S., and Graham, A. (1995), "Migrating to object technology", Addison-Wesley Longman Publishing Co., Inc.

Guizzardi, G., H. Herre, and Wagner, G. (2002), "On the General Ontological Foundation of Conceptual Modeling", *Proceedings of the 21st International Conference on Conceptual Modeling, Springer-Verlag, Berlin.*

Haseman, W., and Whinston, A. (1976), "Design of a Multidimensional Accounting System." *Accounting Review* Vol. 51 No. 1, pp. 65-79.

Hay, D. (1999), “A Comparison of Data Modeling Techniques.” *Essential Strategies Vol. 41, pp. 41-93.*

Holland, I. M. and Lieberherr, K. J. (1996), “Object-oriented design”, *ACM Computing Surveys (CSUR)* Vol. 28 No. 1, pp. 273-275.

Ijiri, Y. (1975), “Theory of accounting measurement” (No. 10) American Accounting Association, pp. 3-11.

Kandelin, N.A. and Lin, T. W. (1992), “A computational model of an events-based object-oriented accounting information system for inventory management”, *Journal of Information Systems* Vol. 6 No. 1, pp. 47-62.

Kim, W. (1990), “Object-oriented databases: Definition and research directions”, *IEEE Transactions on Knowledge and Data Engineering* Vol. 2 No. 3, pp. 327-341.

Khoshafian, S. (1993), “Object-oriented databases”, John Wiley & Sons, Inc.

Khoshafian, S. and Abnous, R. (1990), “Object orientation: concepts, languages, databases, user interfaces”, John Wiley & Sons, Inc..

Lanza, R., (1998), “Take My Manual Audit, Please” *Journal of Accountancy* Vol. June, pp. 33-36

Larkin, J. H. and Simon, H. A. (1987), “Why a diagram is (sometimes) worth ten thousand words”, *Cognitive science* Vol. 11 No. 1, pp. 65-100.

Lieberman, A. Z., and Whinston, A. B. (1975), “A Structuring of an Events-Accounting Information System.” *The Accounting Review* Vol. 50 No.2, pp. 246-258.

Lorenz, M. (1993), “Object-Oriented Software Development: A Practical Guide”, Prentice-Hall, Inc.

Maier, D. and Stein, J. (1987), “Development and implementation of an object-oriented DBMS, Research directions in object-oriented programming”, MIT Press Cambridge, MA, USA.

Motro, A. (1989), “Integrity = validity + completeness.” *ACM Transactions on Database Systems* Vol. 14 No. 4, pp. 480-502.

Mattessich, R., (1964), “Accounting and Analytical Methods”, Homewood, Illinois: Richard D. Irwin.

Odell, J. J. and Martin, J. (1995), “Object Oriented Method: A Foundation”, Prentice-Hall Englewood Cliffs.

McCarthy, W. (1979), “An entity-relationship view of accounting models.” *Accounting Review* Vol. 54 No. 4, pp. 667-686.

McCarthy, W. (1982), “The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment.” *Accounting Review* Vol. 57 No. 3, pp. 554-578.

McClure, S. (1997), “Object Database vs. Object-Relational Databases.” *IDC Bulletin #14821E (*August).

Morris, M. C. Speier, and Hoffer, J. (1999), “An Examination of Procedural and Object-oriented Systems Analysis Methods: Does Prior Experience Help or Hinder Performance?” *Decision Sciences.* Vol. 30, pp. 107-136.

Murthy, U. S., and Wiggins, C. E. (1993), “Object-oriented approaches for designing accounting information systems.” *Journal of Information Systems* Vol. 7 No. 2, pp. 97-111.

Murthy, U. S., and Wiggins, C. E. (1997), “Event Focused Object Modeling: A Methodology for Designing Object-Oriented Business Information Systems” *Working Paper.*

Nagarur, N. N., and Kaewplang, J. (1999), “An object-oriented decision support system for maintenance management.” *Journal of Quality in Maintenance Engineering* Vol. 5, No. 3, pp. 248-257.

Nakamura, H., and Johnson, R. (1998), “Adaptive Framework for the REA Accounting Model”, *OOPSLA'98 Workshop on Business, Object Design and Implementation IV.*

Navathe, S. (1992), “Evolution of data modeling for databases”, *Communications of the ACM* Vol. 35 No. 9, pp. 112-123.

Nelson, T. H. (1965), “Complex Information Processing: A File Structure for the Complex, the Changing and the Indeterminate”, ACM '65: *Proceedings of the 1965 20th national conference*, pp. 84 – 100.

Newell, A. and Simon, H. A. (1972), “Human Problem Solving”, Englewood Cliffs, NJ: Prentice-Hall.

O'Keefe, R. (1986), "Simulation and Expert Systems: A taxonomy and Some Examples", *Simulation* Vol. 46 No. 1, pp.10-16.

Orman, L. V. (1997), "Database Auditing." *Americas Conference on Information Systems.*

Rob, C., and Coronel, C. (1999), "Database Systems Design, Implementation, and Management", Thomas Learning.

Rolland, C., Proix, C. (1992), "Natural Language Approach to Conceptual Modeling" in *Conceptual Modeling, Databases and CASE: An Integrated View of Information Systems Development,* WILEY.

Shoval, P., and Shiran, S. (1997), "Entity-relationship and object-oriented data modeling – an experimental comparison of design quality", *Data & Knowledge Engineering* Vol. 21, pp. 297-315.

Smelcer, J., and Carmel, E. (1997), "The Effectiveness of Different Representations for Managerial Problem Solving: Comparing Tables and Maps", *Decision Sciences* Vol. 28 No. 2, pp. 391-420.

Smith, J. M. and Smith, D. C. (1977), "Database Abstractions: Aggregation and Generalization", *ACM Transactions on Database Systems (TODS)* Vol. 2 no. 2, pp. 105-133.

Smith, K. E., and Zdonik, S. B. (1987), "Intermedia: A Case Study of the Difference between Relational and Object-Oriented Database Systems." *OOPSLA '87 Proceedings.*

Sorter, G. (1969), "An "Events" Approach to Basic Accounting Theory." *Accounting Review* Vol. 44 No. 1, pp. 12-19.

Taivalsaari, A. (1996), "On the notion of inheritance", *ACM Computing Surveys (CSUR)*, Vol. 28 No. 3, pp. 438 – 479.

Vessey, I. (1991), "Cognitive Fit: A Theory-Based Analysis of Graph versus Tables Literature." *Decision Sciences* Vol. 22, pp. 219-240.

Wang, S. (1996), "Towards Formalized Object-Oriented Management Information Systems Analysis." *Journal of Management Information Systems*. Vol. 12, No. 4, pp. 117-141.

Yu, S. C., (1976), "The Structure of Accounting Theory", University Presses of Florida.