A COMPARISON OF DIFFERENT APPROACHES TO MODELING FINANCIAL STATEMENTS

Grace O'Farrell*, Chunhui Liu**

* Faculty of Business and Economics, University of Winnipeg, Winnipeg, Manitoba, Canada
** Caritas Institute of Higher Education, Hong Kong

Abstract

This paper describes the relational, entity-relationship (ER), and object-based approaches to modeling financial statements; and discusses the strengths, weaknesses, and user adaptability of these models. We believe that the relational, ER, and object-oriented models may not be individually adequate to model the accounting processes in an integrative accounting information system. The increasing amount of disclosures in the footnotes to the financial statements and the complex compliance requirements of the Sarbanes-Oxley Act suggest that the object-relational model may be appropriate to model both the quantitative and qualitative items in the accounting processes. The object-relational model builds on the strengths of the relational, ER, and object-oriented models and mitigates the weaknesses of these models. We develop a set of propositions based on our review of the current literature on the conceptual models.

Keywords: Financial Statement Modeling, Object-Oriented Model, Object-Relational Model, Entity-Relationship Model, Relational Model, Services and Standards

Acknowledgment

This paper is dedicated to the memory of Professor Lee J. Yao.

1. INTRODUCTION

The objective of the accounting function is to provide information to users for decision-making. Accounting information systems have been designed and developed to help accountants identify, measure, and communicate information to users (Williams et al., 2015). While novice users might be overwhelmed by the availability of information currently generated by the system, sophisticated users may demand more detailed information at a disaggregated level so that they can aggregate the data in a manner that they deem fit for enhanced decision-making. The demand for increased information is made possible by the database approach to accounting. The underlying framework for the database accounting system is Sorter's events theory. Events theory proposes collection of data in a disaggregated form for aggregation by users based on their individual decision models. This means that users are allowed access to the disaggregated data stored in the system. That is, instead of providing users with a sales figure in the traditional income statement, the user can access a database and retrieve all the attributes of a firm's business transaction (i.e., date of sale, amount of sale, sales discount, terms of sale, warranties, control procedures for credit check or credit approval, control procedures for account write-off, etc.) from the system and aggregate the data based on their decision models. The entity-relationship (ER) model (Chen, 1976) is a widely used conceptual tool for implementing events theory and incorporating this implementation in the design of accounting information systems.

Research shows that new standards have significant implications to existing services (Hossain 2014; Liu and O'Farrell, 2013a; Mendoza and Ravichandran, 2013). The Sarbanes-Oxley Act (SOX) of 2002 has increased the responsibilities of management in the establishment and maintenance of an adequate internal control structure and appropriate control procedures. Management is required to report this information in the accompanying financial statements. Such a requirement calls for an expansion of the current accounting information system to include not only the quantitative information contained in the traditional financial statements, but also the increasing amount of qualitative information required to comply with the reporting requirements of SOX and the increasing amount of disclosures in the footnotes to the traditional financial statements. The qualitative rather than quantitative nature of such information makes the task of assessing a model more subjective and less reliable (Liu and O'Farrell, 2013b). Thus, a major problem of conceptual models lies in their evaluation. Users might interpret the conceptual models differently based on their knowledge of a specific domain and their perception of various objectives (Dunn et al., 2005; Liu and O'Farrell, 2013b).

The existing conceptual models – relational, ER, and object-oriented approaches may not be adequate individually for modeling the accounting processes in an integrative accounting information system. Since the object-relational model builds on the strengths and mitigates the weaknesses of the relational, ER, and object-oriented approaches, we posit that the object-relational model is appropriate for modeling accounting processes that possess both quantitative and qualitative characteristics. We extend the object-relational model to include not only the quantitative information contained in the financial statements, but also the qualitative information required to comply with the reporting requirements of SOX and the disclosures in the footnotes to the financial statements. . This is important for financial reporting, disclosure and transparency of accounting information to users. We choose to model a specific accounting process such as a sales cycle because of its inherent characteristics such as the criticality of the sales function in any business, voluminous and repetitive nature of sales, abundance of less easily defined parameters (i.e., control procedures for credit check or credit approval, control procedures for account write-off, etc.), and technological feasibility of modeling sales on a real-time basis. In addition, the quantitative and qualitative characteristics of a sales model are programmable and testable. The relatively similar structure of the financial statements of most firms suggests that it might be possible to use a generic conceptual model to portray the accounting information system of a firm.

This paper consists of five sections. The next section describes the relational, entity relationship (ER), object-oriented, and object-relational models. Section III discusses the strengths, weaknesses, and user adaptability of the models. Section IV presents a set of propositions based on our review of the current literature on the models. Section V provides concluding remarks.

2. DESCRIPTION OF THE MODELS

2.1. Relational model

A relational model is a popular tool for organizing data in information systems (Philippi, 2005). A relational model (Codd, 1970) is a set of principles based on relational algebra, calculus, and relational database management. To solve the problem of data redundancy, Codd suggested that data should be represented as a set of tables. Codd used the term "relation" to link the tables. The design of this conceptual model includes development of a set of candidate relations and decomposition of a single comprehensive table into a set of tables to conform to the third or higher normal forms. This approach to database design is known as normalization that searches for dependencies among table elements (Yang, 2003).

We illustrate a situation where the relational model is used to implement a multi-level structure of an accounting information system. At the first level, assets consist of current and noncurrent assets. At the next level, current assets include cash, accounts receivables, current notes receivables, inventories, prepaid expenses, etc. Similarly, noncurrent assets include land, buildings, equipment, long-term notes receivables, intangible assets, other fixed assets, etc. In the relational data model, a single table can be used to represent all current and noncurrent assets or two separate tables can be used to represent the current assets and noncurrent assets respectively. When a single table is used to represent all assets, current assets are not clearly distinguishable from non-current assets. In the case where two separate tables are used to represent current and noncurrent assets respectively, it is essential for users to know that these two tables are related. The types of assets may vary among firms in the same industry or across firms in different industries. To solve this problem, a table can be created for each type of asset or a universal table can be created with a null value in the column where a firm does not have a specific type of asset. A common example of structural variations is the use of footnotes which are common but not attributable to every item in the financial statements. The non-uniformity of a database structure introduces complexities in the formulation of queries.

The relational data approach for modeling the sales cycle is presented below:

Table1. The sales cycle

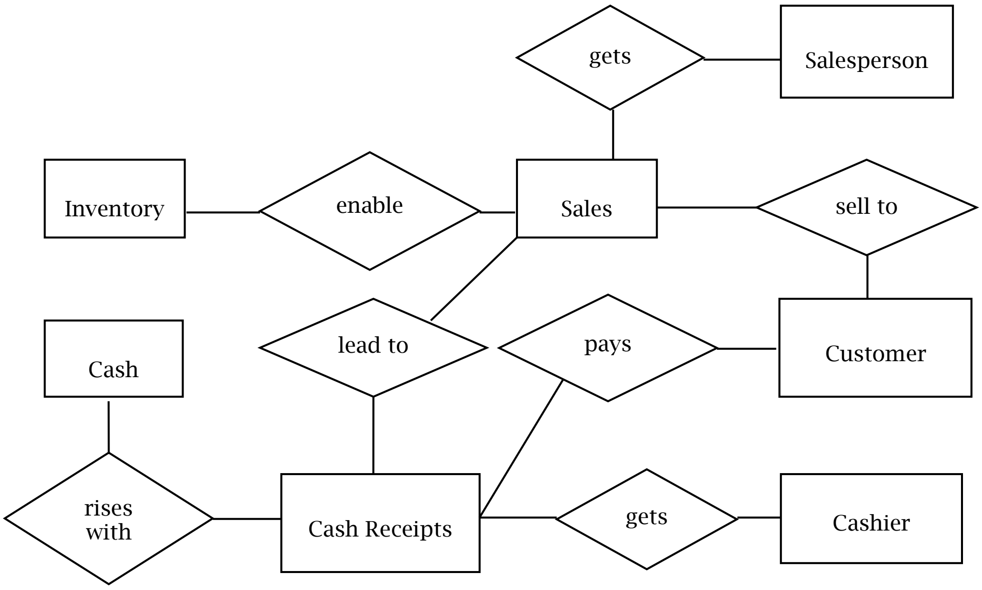
Sales	Invoice number, Invoice Date, Customer number, Employee number, Amount of sale
Cash Receipts	Receipt number, Receipt date, Customer number, Employee number, Receipt amount, Discount applied, Sales tax collected
Cash Receipts_Sales	Receipt number, Invoice number, Amount remitted
Inventory	Item number, Description, Quantity on hand, Unit cost, Unit selling price
Inventory_Sales	Item number, Invoice number, Quantity
Customers	Customer number, Name, Address, Contact number, Credit limit
Employees	Employee number, Name, Address, Contact number, Date of birth, Pay rate

2.2. Entity-relationship (ER) Model

The entity-relationship (ER) model is a conceptual design tool. This model represents the knowledge domain as a collection of basic objects (entities) and provides structure for relationships among these objects (Chen, 1976; Moody and Kortink, 2003; Thalheim, 2000). Unlike the relational model, the ER approach is not based on a formal mathematical model. Instead, the ER model uses a natural way to map real world objects and their relationships directly into database structures. The ER model is based on the notion that an entity can be any discrete object. Two or more entities can be connected via relationships. Both the entities and relationships can have attributes. A graphical representation of entities is a rectangle while relationships are drawn as diamonds. A certain degree of ambiguity might exist in the ER approach because components can be defined in a subjective manner by the designer. Despite the ambiguity problem, simplicity and wide-applicability of the ER model make this approach a popular tool for capturing underlying semantics of business processes (Badia, 2004). Many additions and enhancements have been developed to improve the conceptual design of the ER model. The enhanced ER (EER) model has been proposed to incorporate new concepts of class/subclass relationships, specialization, and generalization (Elmasri and Navathe, 2001). This approach overcomes the weaknesses of semantic models and improves presentation of data properties and constraints. Attribute-oriented business requirements and constraints are developed to improve visualization of business requirements and constraints at the conceptual level (Khan et al., 2004).

An example of an ER model of a sales cycle is presented as follows:

Figure 1. Entity-relationship model of a sales cycle



2.3. Object-oriented Model

The increasing need for information in aggregated or disaggregated form leads to a search for and development of models for presenting accounting information. The object-oriented approach (Martin and Odell, 1992; Murthy and Wiggins, 1993; Roohani and Sutton, 1997) models accounting processes to provide flexibility for a variety of decision models. Unlike entities in the ER model, each object in the object-oriented model has its own unique identity (Kimour and Meslati, 2005). The object-oriented model uses the principles of object-oriented programming. This approach is an expressive modelling technique that supports various semantic primitives in data representation (Liu and Theodor, 2004). The object-oriented model is based on the concepts of objects and relations among the objects. This approach provides flexible structuring capabilities and allows for explicit specification of data constraints. One fundamental idea of the object-oriented approach is the direct mapping between objects and codes (Bézivin, 2005). This model describes the real world using notions such as data, inheritance, reuse and message passing (Zhang, 2001). The object-oriented model is a collection of objects that interact with one another via transmission of messages and maintenance of their own state.

Another fundamental idea of the object-oriented model is encapsulation of data and the code that operates on that data. The object classes are similar to entities in the ER model and attribute values are known as instance variables. An object is a particular instance of an object class. An advantage of containment relationships is that objects can be shared among several containment objects. Objects may be operated via messages from other objects. A message causes an execution of a method, a procedure written in the programming language. A method or code can manipulate the object's local instance variables or send messages to other objects. This encapsulation of data and code is a feature of the object-oriented model. Objects with the same types of values and methods are called classes. Classes can be described as a type definition for objects. The inheritance hierarchy states that each class can inherit the attributes and methods of another class.

The customer class in a sales cycle can be represented as:

Table 2. The customer class in a sales cycle

Variables	**string** name;
	string address;
	string contact_number;
	int customer_number;
	int credit_limit;
	int balance
Messages	**string** get-name();
	string get-address();
	int set-address (**string** new-address);
	int balance(**date** current-date)

The customer class has the following objects: name, address, contact_number, customer_number, credit_limit, and balance. The methods for retrieving the messages are as follows: get-name(), get-address(), set-address(new-address), and balance(current-date).

According to the inheritance hierarchy, the complete name for an inventory object is financial_statement.balance_sheet.current_assets.inv entory. The inventory class in a sales cycle can be represented as:

Table 3. The inventory class in a sales cycle

Variables	**string** description;
	int item_number;
	int quantity_on_hand;
	int unit_cost;
	int unit_selling_price
Messages	**string** get-description();
	int get-item_number();
	int set item_number (**int** number);
	int set unit_cost (**int** invoice.unit_cost(**int** number));
	int get cogs(**int** quantity);
	int get unit_selling_price();
	int get sales(**int** invoice.quantity(**int** number));
	int get balance(**date** current-date)

The inventory class has the following objects: description, item_number, quantity_on_hand, unit_cost, and unit_selling_price. The methods for retrieving the messages include: get-description(), get item_number(), set item_number (number), set unit_cost (invoice.unit_cost(number)), get cogs(quantity), get unit_selling_price(), get sales(int invoice.quantity(number)), get balance(current-date). These messages facilitate the setting and retrieval of item number, unit cost, unit selling price; and computation of cost of goods sold, the amount received, and the balance.

2.4. Object-relational Model

The characteristics of effective conceptual modeling include: (1) an expressive model that distinguishes different data types, relationships, and constraints; (2) a simple model that is easy to understand; (3) a model with minimal, distinct, and orthogonal basic concepts that can be defined in a formal manner; and, (4) a model with equivalent interpretation (Navathe, 1992). The object-relational model, a hybrid of the ER and object-oriented models, retains the relational concepts and extends modelling power by combining the semantic data model of ER and encapsulation of data and code in object-oriented modelling (Silberschatz et al., 2002; Stonebraker and Brown, 1999). The basic relational model is extended by adding the concepts of objects and constructs to the relational query languages to preserve declarative access to data and to mitigate the limitation of atomic domains inherent in the relational approach (Silberschatz et al., 2002). The object-relational model allows for complex types such as nested relations and inheritance at two levels: attribute domains and relations (Silberschatz et al., 2002).

When the object-relational approach is used to model a sales cycle, the objects - Customer, Inventory, Terms and Invoice - can be represented in Table 4.

The first three statements define the types - Customer, Inventory and Terms. The fourth statement defines the Invoice structure type that uses the Customer, Inventory and Terms types. The last statement defines an object table.

Table 4. The objects in a sales cycle

Create type Customer as object	name varchar(20),
	address varchar(20),
	contact_number varchar(20),
	customer_number integer,
	credit_limit integer,
	balance integer
Create type Inventory as object	description varchar(20),
	item_number varchar(20),
	quantity integer,
	unit_cost integer,
	unit_selling_price integer)
Create type Terms as object	net_due integer,
	discount_percentage integer,
	discount_days integer)
Create type Invoice as object	number integer,
	inventory_customer Customer,
	address varchar(20),
	invoice_date date,
	terms_inv Terms,
	inv_item Inventory
Create table Invoices of Invoice	

3. STRENGTHS, WEAKNESSES, AND USER ADAPTABILITY OF THE MODELS

Table 5 provides an overview of the strengths, weaknesses, and user adaptability of the relational, ER, object-oriented, and object-relational models.

3.1. Advantages and Caveats of a Relational Model

The relational model has some advantages (Wilfred, 2001; Navathe, 1992). One advantage is that data can be organized into tables to provide data integrity, eliminate redundant data, and reduce data inconsistencies. A second advantage is that the model can be designed to allow for easy and accurate retrieval of information so that data from more than one table can be used to answer the user's query. However, the relational model has several limitations (Navathe, 2003; Osei-Bryson and Ngwenyama, 2004;Yang, 2003). First, each data entry in a table must be of an atomic type (e.g., numeric or alphanumeric) (Elmasri and Navathe, 2004). Second, the expressive power of the model is limited in terms of portrayal of multi-level items. Third, the relational model is not designed for knowledge representation. Fourth, a user needs to know the structural details of the data before a query is performed. Finally, the relational model is not flexible because it is difficult to use the model to represent different variations of financial statements. Thus, changes to the model structure pose a challenge to user adaptability.

Table 5. A Comparison of the Strengths, Weaknesses, and User Adaptability of the Models

Model	Strengths	Weaknesses	User Adaptability
Relational	- Data integrity - Elimination of redundant data - Reduction of data inconsistencies - Easy and accurate retrieval of information from more than one table to answer user's query	- Limitation of data type (i.e., atomic) - Limited expressive power - Lack of knowledge representation - User knowledge of the structural details of the data - Lack of flexibility	Changes to the structure of the model pose a challenge to user adaptability
ER	- Simple graphical representations - Strict hierarchic structure - Flexibility - Ease of use	- General approach to modeling such as automating the existing manual accounting systems and not altering the flow of information - Limited expressive power - No associated query language	User adaptability is improved if changes are made within the design of the model
Object-oriented	- Use of complex data types - Extensibility - Inheritance - Flexible structuring capabilities and explicit specification of data constraints - Direct mapping between models and objects - Incorporation of internal controls into the system and provides an audit trail of activities	- Lack of well-formulated rules - Lack of a declarative language - Lack of a generic way for accessing complex objects	Ease of adaptability to changes by a knowledgeable user
Object-relational	- Builds on the strengths of the relational and object-oriented models - No limitation of atomic domains - Declarative access to data - Allows for modeling of quantitative and qualitative items in an accounting cycle	- Vague semantics	Adaptation requires knowledge of the model

3.2. Advantages and Caveats of an Entity-relationship (ER) Model

The strengths of the ER model include use of simple graphical representations, strict hierarchic structure, flexibility, and ease of use. Thus, user adaptability is enhanced if changes are made within the structure of the model. The weaknesses of the ER approach are inherent in a conceptual model and most of the limitations are rooted in the general approach to modeling such as automating the existing manual accounting systems and not altering the flow of information (Elmasri et al., 1985; Hale and Sutton, 1990; Roohani and Sutton, 1997). The ER model lacks expressive power and does not have an associated query language (Markowitz and Shoshani, 1992; Shankant and Navathe, 1992; Dey et al., 1999).

3.3. Advantages and Caveats of an Object-oriented Model

The object-oriented model has several advantages (Roohani and Sutton, 1997). First, the object-oriented model incorporates complex data types and enhances flexibility in organizing the data into columns in the tables. For example, we can use columns that contain collections of values of a single type and columns that contain multiple types, and compare these with the tables contained in the relational model. Second, the object-oriented model uses new data types, access methods, and functions for support (extensibility). New data types can be defined by combining one or more existing data types. Data in the form of numeric, text, video, image, and sound can be embedded in abstract data types. Third, the object-oriented approach defines objects (types and tables) that inherit the properties of other objects and add new properties specific to that object. This concept is known as inheritance. The object-oriented codes can be reused to extend existing applications or develop new applications because the inheritance type facilitates the use or extension of existing object types. Fourth, the object-oriented approach provides flexible structuring capabilities and allows for explicit specification of data constraints. Fifth, the object-oriented approach allows for direct mapping between the model and the objects (Bézivin, 2005). Finally, since users' requests are determined by operations, an object-oriented model opens up opportunities for incorporation of internal controls into the system and provides an audit trail of activities performed by employees. A knowledgeable user can easily adapt to changes in the structure of the model. However, the object-oriented model has weaknesses such as the lack of well-formulated rules, the lack of a declarative language which leads to unavailability of the model to novice users, and the ask of a generic way for accessing complex objects.

3.4. Advantages and Caveats of an Object-relational Model

The object-relational model builds on the strengths of the object-oriented and relational models and mitigates the weaknesses of these models. In addition, the object-relational model does not have the limitation of atomic domains and allows declarative access to data. This approach is appropriate for modelling both quantitative and qualitative items in an accounting cycle. User adaptability requires knowledge of the model. Vague semantics is a weakness of this approach.

4. PROPOSITIONS

We suggest the following propositions based on our review of the current literature on the four conceptual models:

Proposition 1: The results of modeling a sales cycle can be generalized to other accounting cycles in an integrative accounting information system.

Proposition 2: Use of a specific cycle for testing will enhance the applicability and comparability of the results obtained to other accounting cycles.

Proposition 3: The object-relational model is appropriate for modeling both quantitative and qualitative items in an accounting cycle.

5. CONCLUSION

This paper describes the relational, ER, object-oriented, and object-relational approaches to modeling financial statements, and discusses the strengths, weaknesses, and user adaptability of these models. The increasing amount of disclosures in the footnotes to financial statements and the complex compliance requirements of SOX, call for a modeling technique that captures both the quantitative and qualitative aspects of the accounting processes. . The relational, ER, and object-oriented approaches may not be adequate individually for modeling the increasing amount of quantitative and qualitative items in financial statements. . This is important for financial reporting, disclosure and transparency of accounting information to users. Since the object-relational approach builds on the strengths of the relational, ER, and object-oriented approaches and mitigates the weaknesses of these models, the object-relational approach may be ideal for modeling both quantitative and qualitative characteristics of the accounting processes. Future research can test and compare the relational, ER, object-oriented, and object-relational models to provide empirical evidence on the appropriateness of these approaches for modeling quantitative and qualitative characteristics of the accounting processes in an integrative accounting information system. Specifically, future work can test the propositions developed in this paper to enhance understanding of modeling techniques.

REFERENCES

1. Badia, A. (2004), "Entity-relationship modeling revisited", SIGMOD record, Vol. 33, No.1, pp. 77-82.
2. Bézivin, J. (2005), "On the unification power of models", Software and Systems Modelling, Vol. 4, No. 2, p. 171.
3. Chen, P.P. (1976), "The entity-relationship model: toward a unified view of data", ACM transactions on database systems, Vol. 1, pp. 9-36.
4. Codd, E.F. (1970), "A relational model for large shared data banks", Communications of the ACM, Vol. 13, No. 6, pp. 377-387.
5. Dey, D., Storey, V.C., and Barron, T.M. (1999), "Improving database design through the analysis of relationships", ACM transactions on database systems, Vol. 24, No. 4, pp. 453-486.
6. Dunn, C.L., Gerard, G.J., and Grabski, S.V. (2005), "Critical evaluation of conceptual data models", International Journal of Accounting Information Systems, Vol. 6, pp. 83-106.
7. Dunn, C.L., and McCarthy, W.E. (1997), "The REA accounting model: Intellectual heritage and prospects for progress", Journal of Information Systems, Vol. Spring, pp. 31–51.
8. Elmasri, R. and Navathe, S. (2001), Fundamentals of database systems. Benjamin-Cummings Publishing Co., Redwood City, CA.
9. Elmasri, R. and Navathe, S. (2004). Fundamentals of database systems, 4th Ed. Pearson Education, Inc., Boston, MA.
10. Elmasri, R., Weeldreyer, J., and Hevner, A. (1985), "The category concept: an extension to the entity-relationship model", Data and knowledge engineering, Vol. 1, No. 1, pp. 75-116.
11. Everest, G. and Weber, R. (1977), "A relational approach to accounting models", The Accounting Review, Vol. 52, Spring, pp. 340-359.
12. Geerts, G.L and McCarthy, W.E. (1992), "The extended use of intensional reasoning and epistemologically adequate representations in knowledge-based accounting systems", Proceedings of the 12th International Workshop on Expert Systems and Their Applications, Avignon, France, pp. 321–332.
13. Hale, D.P. and Sutton, S.G. (1990), "Integrating semantic and temporal dimensions into accounting models", American Accounting Association annual meeting, p. 32.
14. Hossain, M.M. (2014), "Pervasiveness of SERVQUAL and its potential for the standards for functional quality of service", International Journal of Services and Standards, Vol. 9, No. 1, pp. 67-83.
15. Khan, K.M., Kapurubandara, M., and Chadha, U. (2004), "Incorporating business requirements and constraints in database conceptual models", Conferences in Research and Practice in Information Technology, Vol. 31, pp. 59-64.
16. Kimour, M. and Meslati, D. (2005), "Deriving objects from use cases in real-time embedded systems", Information and Software Technology, Vol. 47, No. 8, p. 533.
17. Lampe, J.C. (2002), "Discussion of an ontological analysis of the economic primitives of the extended-REA enterprise information architecture", International Journal of Accounting Information Systems, Vol. 3, pp. 17-34.
18. Lieberman, A.Z. and Whinston, A.B. (1975), "A structuring of an events-accounting information system", The Accounting Review, April, pp. 246-258.
19. Liu, C. and O'Farrell, G. (2013a), "The impact of XBRL on forecast accuracy across nations", International Journal of Services and Standards, Vol. 8, No. 3, pp. 247-263.
20. Liu, C. and O'Farrell, G. (2013b), "The role of accounting values in the relation between XBRL and forecast accuracy", International Journal of Accounting and Information Management, Special Issue on Behavioral Finance and Accounting, Vol. 21, No. 4, pp. 297-313.
21. Liu, D. and Theodor, S. (2004), "Integrated object-oriented framework for MCDM and DSS modeling", Decision Support Systems, Amsterdam: Dec 2004, Vol. 38, No. 3, p. 421.
22. Markowitz, V.M. and Shoshani, A. (1992), "Representing extended entity-relationship structures in relational databases: a modular approach", ACM transactions on database systems, Vol. 17, No. 3, pp. 423-464.
23. Martin, J. and Odell, J.J. (1992), Object-oriented analysis and design, Prentice Hall, Englewood Cliffs, New Jersey.
24. McCarthy, W. (1982), "The REA accounting model: a generalized framework for accounting systems in a shared data environment", The Accounting Review, July, pp. 554-578.
25. Mendoza, R.A. and Ravichandran, T. (2013), "Organizational assimilation of vertical standards: exploring the interplay of technology destiny, firm-level factors and network effects", International Journal of Services and Standards, Vol. 8, No. 1, pp. 1-23.
26. Moody, D. and Kortink, M. (2003), "From ER Models to Dimensional Models, Part II: Advanced

Design Issues", Journal of Data Warehousing, Vol. 8, No. 4, p. 20.
27. Moody, D.L., Sindre, G., Brasethvik, T., and Solvberg, A. (2003), "Evaluating the quality of information models: empirical testing of a conceptual model quality framework", Proceedings of the 25th international conference on Software engineering, Portland, Oregon, pp. 295-305.
28. Murthy, U.S. and Wiggins, C.E. Jr. (1993), "Object-oriented approaches for designing accounting information systems", Journal of Information Systems, Fall, pp. 97-111.
29. Navathe, S.B. (1992), "Evolution of data modeling for databases", Communications of the ACM, Vol. 35, No. 9, pp. 112-123.
30. Osei-Bryson, K.M. and Ngwenyama, O. (2004), "Supporting Semantic Diversity in the Relational Model: The Case of Multi-Faced Attributes", Information Systems Frontiers, pp. 277-285.
31. Philippi, S. (2005), "Model driven generation and testing of object-relational mappings". The Journal of Systems and Software. Vol. 77, No. 2, p. 193.
32. Roohani, S.J. and Sutton, S.G. (1997), "An object-oriented approach to the design of real-time accounting information systems", Accounting Forum, Vol. 20, Vol. 3-4, pp. 293-310.
33. Shanks, G., Thansley, E., and Weber, R. (2003), "Using ontology to validate conceptual models", Communications of the ACM, Vol. 46, pp. 85-89.
34. Stonebraker, M and Brown, P. (1999), Object-relational DBMSs tracking the next great wave, Morgan Kaufmann Publishers, Inc., San Francisco, CA.
35. Thalheim, B. (2000), Entity-Relationship Modeling: Foundations of Database Technology, Springer-Verlag, Berlin Heidelberg.
36. Wilfred, N.G. (2001), "An extension of the relational data model to incorporate ordered domains", ACM transactions on database systems, Vol. 26, No. 3, pp. 344-383.
37. Williams, J.R., Haka, S.F., Bettner, M.S., and Carcello, J.V. (2015), Financial & Managerial Accounting, McGraw-Hill Education, New York, NY.
38. Yang, Heng-Li. (2003), "Comparing relational database designing approaches: Some managerial implications for database training", Industrial Management & Data Systems, Vol. 103, No. 3-4, pp. 150-167.
39. Zhang, Q. (2001), "Object-oriented database systems in manufacturing: Selection and applications", Industrial Management and Data Systems, Vol. 101, No. 3, pp. 97-105.